2D Crystallization and Image Processing of Membrane Proteins

Inauguraldissertation
zur
Erlangung der Würde eines Doktors der Philosophie
vorgelegt der
Philosophisch-Naturwissenschaftlichen Fakultät
der Universität Basel

von

Gian Andrea Signorell

aus Zuoz GR

Basel, 2008

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät
auf Antrag von

Prof. Dr. Andreas Engel und Prof. Dr. Tilman Schirmer

Basel, den 27.10.2008

Prof. Dr. Hans-Peter Hauri
Dekan

Contents

List of Figures

Acknowledgements

I wish to thank Ansgar Philippsen for the support and the collaboration in the software related part of my thesis. I'm thankful for the great collaboration and advice of Andreas Schenk and the rest of the IPLT team (Valerio Mariani, Simon Berneche and Johan Hebert).

Thanks goes also to the wet-lab team supporting me in my MBCD and crystallization experiments. Here Hervé Remigy, Mohamed Chami, Wanda Kukulski and Thomas Kaufmann were the persons supporting and collaborating.

The KdgM, KdgN and NanC protein was provided by Guy Condemine, the AQP1 and OmpF protein was produced by Kitaru Suda and the SoPiP2;1 by Wanda Kukulski.

Thanks to Patrick Frederix for introducing me to the AFM technique.

I especially would like to thank Andreas Engel who gave me the opportunity to work in his laboratory on all these very interesting projects.

Chapter 1

Introduction

The goal of structural biology is to determine the structure of biological molecules such as proteins, lipids, DNA or bigger complexes consisting of these basic building blocks. The determination of structure is an important step to gain insight into the physical and biological functioning of these molecules and complexes. Structure, in the general sense of physical and chemical composition, determines lastly the function.

Several techniques are known and used in the field of structure determination. X-rays can be used to determine structures of crystallized biological molecules or complexes to a high precision, as high as atomic resolution. The drawback of the structure determination using the X-ray technique is that the molecules are not anymore in their biological environment and that they are forced in the crystal packing to conformations they would not adapt in nature. NMR spectroscopy is an other method used to determine the structure of biological molecules at atomic resolution. This method is preferentially used for soluble compounds. The structure of larger complexes could recently be analyzed with both X-ray crystallography (ribosomes [45]) and NMR spectroscopy (GroEL GroES complex [14]), but the determination of such big complexes remains a very difficult issue.

Single particle analysis of electron microscopy images on the other side allows the structural determination of bigger complexes. The drawback of this method is that a much lower level of detail (resolution) can be achieved.

Every technique has its field of application, its advantages but also its drawbacks. It is often the problem one wants to solve, that determines the the way to solve it. Bigger complexes of molecular machines are investigated using the single particle method, small soluble molecules using NMR or membrane bound proteins are investigated using electron crystallography.

Membrane proteins are important in cells, as they can select what passes through a membrane and therefore what moves in or out of a cell or an organelle. Since membrane proteins have a hydrophobic surface, they are stabilized to a great extent by the membrane they are in. The isolated protein is rather unstable and easily loses its functionality when removed from the lipid bilayer. This is the main reason why membrane proteins are so difficult to crystallize in 3D in order to be analyzed with the X-ray crystallography and why they are too unstable in solution for the analysis with NMR spectroscopy. These limitations explain the observation, that only very few structures of membrane proteins are resolved to a high resolution until now.

The generation of 2D crystals and the subsequent analysis with the electron microscope using images or direct electron diffraction is a another way to get structural data of membrane proteins up to a very high resolution as it has been demonstrated by solving the structure of AQP-1 to a resolution of 3.8 Å [34] and then 3.2 Å [10] even before the structure was solved by X-ray crystallography [48]. More recently the structure of AQP-0 was solved by electron crystallography to 1.9 Å, revealing not only the protein, but also the lipids surrounding it [18]. Hence, the big advantage of this method is that the proteins are in a lipid bilayer and therefore in a close to native environment. However, its drawback is that only few steps in the whole process of getting structural information from the purified protein are automated. A systematic screening for ideal crystallization conditions is mandatory but time consuming. The image acquisition with the electron microscope demands highly skilled and experienced users for the sample preparation and the microscope handling in order to get high resolution data. Image analysis and data processing as last step in the structure determination process is of great importance as the information present in the images or in the diffraction data needs to be extracted and interpreted. The advance of data processing was slow for a long time and could hardly cope with the enormous amount of data generated. Only recently more resources are available for developments in this field ([17], [38] and [37]).

Advances in all these presented fields, from protein production up to image processing, are pushed by our group. In this thesis I will present my contributions to the development of a novel method for 2D crystallization, the production of 2D crystals and the development of tools for image processing.

The methods used, the biological background and the insights gained from the performed experiments will be described and discussed. An overview and some insights gained through collaborative work in an interdisciplinary team will also be given.

Chapter 2

2D Crystallization of Membrane Proteins

2.1 Biological Background

Biological membranes are of great importance for all live forms. They are the separators of biological spaces and can therefore confine regions with different chemical properties. The interaction and the exchange of different compounds between the confined spaces needs to be regulated in a strongly controlled way to keep the concentration gradients of the compounds at the desired levels.

In this context membrane proteins play an important role as ’gatekeepers’. As channels they can be selective towards different compounds as big as long sugar chains (KdgM[5]) and iron complexes (HasAR[27]) or as small as water or ions (Aquaporins[15, 52]). These channels control and mediate the traffic of big quantities of some compounds (water for example) and very small quantities of other compounds.

2.1.1 Outer Membrane Porins

2.1.1.1 General Introduction

Gram-negative bacteria are characterized by the composition of their cell wall: The cytoplasmic or inner membrane engulfs the cytoplasm, a murein layer lends mechanical stability and shape. A second, outer membrane surrounds the cell with few contacts to the inner membrane. In between is the periplasm; in this compartment there are some metabolic activities which would disturb the metabolism within the cell - for example reactions dealing with toxic substances. The exchange of substances between the cytoplasm and the periplasm is regulated by highly specific transport systems. The exchange between periplasm and environment occurs via porins, which may be unspecific or specific for groups of substances.

The expression of porins is regulated (the total protein content of the outer membrane is constant). In *E. coli* cells grown in media with low osmotic pressure the nonspecific porin (outer membrane protein) OmpF predominates, at higher osmolarity more OmpC is inserted into the membrane[19]. Lack of phosphate induces the synthesis and insertion of PhoE [24]. Other specific porins are for example LamB for the transport of sugar [42] and Tsx for the transport of nucleosides([22] and [55]). All known porins share the same general composition, independent of their specificity: a chain of 260 - 420 amino acids folds to an anti-parallel β-barrel of 14 to 22 strands.

Function: Porins are tubes with a diameter of about 1 nm which are filled with water. Nonspecific porins allow the diffusion of ions and molecules up to a molecular weight of 600 Da. The diffusion speed depends on both the difference of concentration between the periplasm and extra-cellular bulk, and the molecular weight of the solute.

The passing of ions may be regulated electrically. If (*in vitro*) a voltage of + or - 100 mV is applied, the channel is closed for ions (voltage gating). This phenomenon is also found in another class of beta-barrel pores, the toxins. The mechanism of channel closure is controversial: AFM images of 2D OmpF crystals show steric movements of the loops [33], whereas from Brownian dynamics simulations it was concluded that there is no movement of the loop constricting the inside of the channel [36] and [43]. Probably the applied voltage changes the electrostatic properties of the interior wall. A physiological function of voltage gating is not obvious.

2.1.1.2 OmpF

The outer membrane protein F, better known as OmpF porin, is an integral membrane protein located in the outer membrane of the bacteria, *E. coli*. OmpF porin is a non-specific transport channel that allows for the passive diffusion of small, polar molecules (600-700 Da in size) through the cell's outer membrane. Such molecules include water, ions, glucose, and other nutrients as well as waste products [6]. OmpF porins are found in a trimer conformation within the outer cell membrane. The structure of OmpF is determined and was published in [7] and [6].

Figure 2.1 shows the structure of OmpF. The wall of the pore has a thickness of one amino acid only. On the side of the barrel facing the periplasm the beta strands are connected by short loops or turns. On the other side the loops directed to the environment are larger and variable. The loop connecting beta strands 5 and 6 is of special importance: it is folded into the barrel and constricts the cross section. At the narrowest point there are some ionizable amino acids. The filter properties of the pore are defined at this point. This principle is found in other nonspecific porins too, although there is no sequence homology.

2.1.1.3 KdgM, NanC and KdgN

KdgM (see [4] and[35]) NanC [5] and KdgN belong to a new family of porins [4], which is in turn part of the OmpG super-family. The structure of OmpG has been determined recently [56]. All

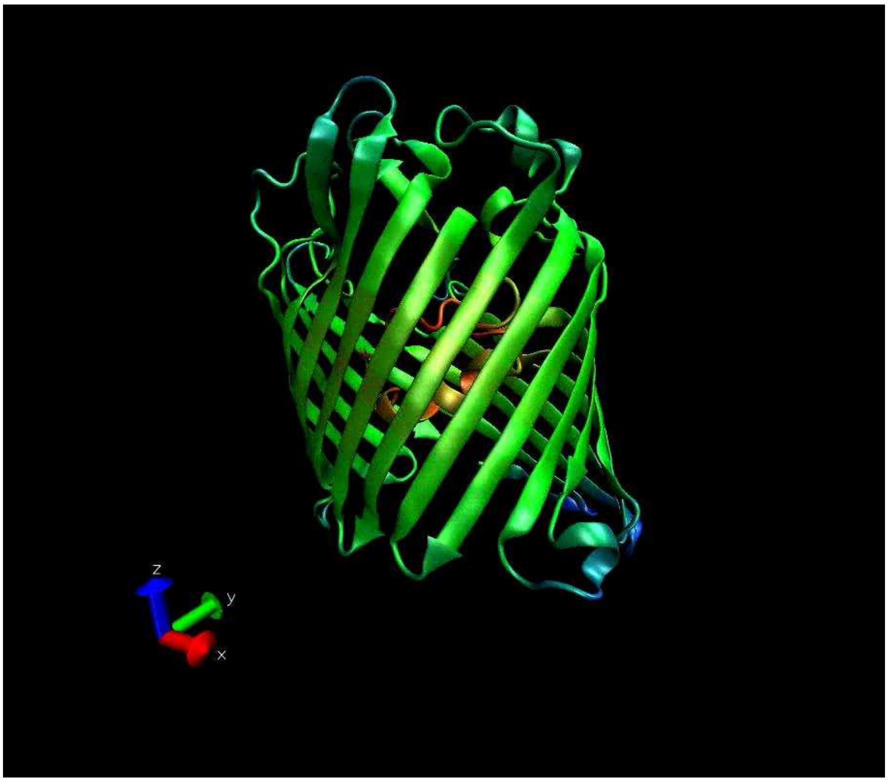


Figure 2.1: Structure of OmpF

The structure of OmpF is presented in ribbon model. The single beta-strands are anti parallel and cross the membrane. The loops facing the environment (top) are larger than those facing the periplasm. The general structure of OmpF is typical for the β-barrel forming porins.

three proteins are substrate selective and behave as high-conductance channels which are open at low membrane potentials and have a weak anion selectivity. The substrates of KdgM and KdgN from *E. chrysanthemi (D. dadantii)* are oligogalacturonides and the substrate of NanC from *E. coli* is sialic acid.

2.1.1.4 HasAR

HasA (heme acquisition system) is an extracellular heme binding protein, a so called hemaphore and HasR a 98 kDa iron-regulated outer membrane receptor protein from *S. marcescens*. They are the initial partners of a heme internalisation system allowing *S. marcescens* (and other bacteria including *P. aeruginosa, Y. pestis* and *Y. enterocolitica* [54]) to scavenge heme at very low concentrations, due to the very high affinity of HasA for heme. Heme is then transferred from HasA to HasR, which has a lower affinity for heme. The mechanism of the heme transfer between HasA and HasR is largely unknown [23].

The crystal structure of HasA has been determined and is shown in [2]. In this paper a hypothetical mechanism for heme uptake and release is suggested.

The specific surface receptor HasR alone is also able to take up free heme or heme from hemoglobin, but the synergy between HasA and HasR greatly facilitates heme uptake allowing hemoglobin uptake at much lower hemoglobin concentrations [16]. Extracellular addition of HasA can facilitate hemoglobin uptake when HasR is present.

2.1.2 Other Membrane Proteins

A number of other membrane proteins were used in my cyclodextrin experiments. They were available from different sources (see section 2.2.5 for details). Following is a short presentation of the biological background of these proteins.

2.1.2.1 Mammalian Porins

AQP1 and AQP8 Aquaporins (AQP) are integral membrane proteins that serve as channels in the transfer of water and, in some cases, small solutes across the membrane. They are conserved in bacteria, plants, and animals. Structural analyses of these proteins have revealed the presence of a pore in the center of each aquaporin molecule. In mammalian cells, more than 10 isoforms (AQP0 - AQP10) have been identified so far (see [49] for a recent review). AQP1 is found in the red blood cells, kidney proximal tubules, eye, and ear and it is claimed that AQP1 forms tetramers in the plasma membrane [51]. AQP1 is the human erythrocyte water channel [28]. AQP8 is expressed in many organs including the kidney, testis, and liver. In addition to its water permeability, AQP8 is permeable to urea but not to glycerol [29].

2.1.2.2 Plant Porins

SoPIP2;1 SoPIP2;1, an aquaporin previously called PM28A, is a plasma membrane intrinsic protein in *S. oleracea* (spinach) leaf plasma membranes. Aquaporins constitute about 20% of all integral membrane proteins in *S. oleracea*. The three dimensional structure was solved and is presented in [50] and [26].

2.1.2.3 Bacterial Membrane Proteins

GalP GalP is an *E. coli* galactose/proton symporter [30]. The substrate specificity of GalP is similar to that of the mammalian glucose transporters [20].

NaChBac NaChBac is a voltage-gated bacterial sodium channel. The ORF of NaChBac has 274 amino acids and its predicted size is 31 kDa. The sequence of NaChBac, especially in the pore region, is similar to that of voltage-gated calcium channels [40].

2.2 Production of 2D Crystals using Cyclodextrin

2.2.1 Foundations and Prior Work

2.2.1.1 Cyclodextrins

Cyclodextrins are non-reducing cyclic glucose oligosaccharides resulting from the cyclomaltodextrin glucanotransferase catalyzed degradation of starch. Their structures have been reviewed in [41]. There are three common cyclodextrins with 6, 7 or 8 D-glucopyranonsyl residues (α-, β-, and γ-cyclodextrin respectively) linked by α-1,4 glycosidic bonds. The glucose residues have the ^{4}C1 (chair) conformation. All three cyclodextrins have similar structures (i.e. bond lengths and orientations) apart from the structural necessities of accommodating a different number of glucose residues. The shape of the cyclodextrin molecule is a bottomless bowl (see Figure 2.2) and the molecule is stiffened by hydrogen bonding between the 3-OH and 2-OH groups around the outer rim. The hydrogen bond strengths are α-cyclodextrin < β-cyclodextrin < γ-cyclodextrin.

The flexible 6-OH hydroxyl groups are also capable to form linking hydrogen bonds around the bottom rim but these are destabilized by dipolar effects and are easily dissociated in aqueous solution. The hydrogen bonding is all 3-OH (donor) and 2-OH (acceptor) in α-cyclodextrin but flips between this and all 3-OH (acceptor) and 2-OH (donor) in β- and γ-cyclodextrins [41].

The cavities have different diameters dependent on the number of glucose units: α-cyclodextrin has an inner diameter of 0.45-0.57 nm and an outer diameter of 1.37 nm; β-cyclodextrin has an inner

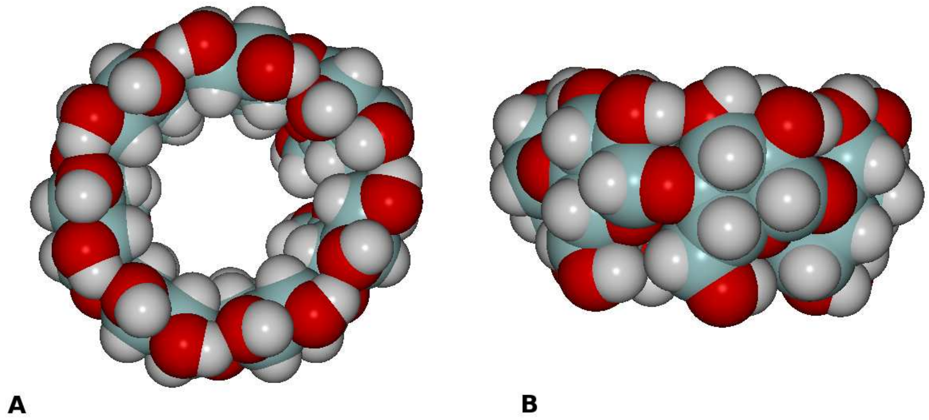


Figure 2.2: β-cyclodextrin
Top view(A) and side view (B) of β-cyclodextrin showing the typical bottomless bowl (truncated cone) shape. The diameter of the ring is different for each of the three cyclodextrin compounds, but the height of 0.8 nm is the same for α-, β- and γ-cyclodextrin. The inclusion compounds (in our experiments the detergent molecules) are kept in the the ring.

diameter of 0.62-0.78 nm and an outer diameter of 1.53 nm; γ-cyclodextrin has an inner diameter of 0.79-0.95 nm and an outer diameter of 1.69 nm. The side rim depth is the same for all three molecules (at about 0.8 nm).

Cyclodextrin rings are amphipathic with the wider rim displaying the 2- and 3-OH groups and the narrower rim displaying 6-OH group on its flexible arm. These hydrophilic groups are on the outside of the molecular cavity whereas the inner surface is hydrophobic lined with the ether-like anomeric oxygen atoms and the C3-H and C5-H hydrogen atoms. In aqueous solution, this hydrophobic cavity contains about 3 (α-cyclodextrins), 7 (β-cyclodextrins) or 9 (γ-cyclodextrins) poorly held (but low entropy) and easily displaceable water molecules. This water in the cavities has low density as the cavities are large enough to accommodate several more molecules. Thus, the otherwise hydrophilic cyclodextrin molecules may bind non-polar suitably-sized aliphatic and aromatic compounds such as lipids.

2.2.1.2 Prior Reconstitution Experiments and First Crystallization Trials

Before we started our experiments using cyclodextrins to generate 2D crystals of membrane proteins, cyclodextrins were used to selectively extract detergent from mixed detergent/lipid/protein micelles in order to produce proteoliposomes. The detergent extraction is based on the much higher affinity of cyclodextrin for detergents than for bilayer-forming lipids. The experiments produced good results with a variety of detergents, independent of their type and critical micelle concentration [11].

Hervé Remigy had the idea to use this method of detergent removal to produce 2D crystals of membrane proteins. The direct addition of highly concentrated cyclodextrin to a ternary lipid-protein-detergent mixture yielded in the aggregation of the protein. The conclusion of this observation was that there was not enough time for the ordering and finally crystalline arrangement of the protein molecules in two dimensions because of the too fast detergent removal.

Slower detergent removal experiments were conducted by adding cyclodextrin at a low concentration (solubilized in nanopure water) over a longer time period to the ternary lipid-protein-detergent mixtures. The slow addition of cyclodextrin solution was performed using a device originally constructed for the production of 2D crystals with dilution [39].

The first experiments of this kind were performed using the proteins SoPIP2;1 and OmpF, since crystallization conditions for this proteins were already known from prior experiments performed with dialysis buttons [26, 13] and since the purified protein was available in reasonably large quantities. The same conditions in terms of salts, LPR and pH that have been used for the successful crystallization experiments using dialysis were used for the cyclodextrin experiments. Since we had no experience about the kinetic behaviour of the crystal formation using cyclodextrins to remove detergent from the ternary mixtures, we performed experiments over a variety of time ranges. The required amount of cyclodextrin was added over time ranges of 30 minutes, 2 hours, 12 hours and 72 hours. Additional experiments over a time range of 16 hours were performed with SoPIP2;1.

SoPIP2;1 crystallized only in the 72 hours and 16 hours experiments. No crystals could be found in the shorter experiments. In contrast to the other experiments, OmpF yielded crystalline vesicles in the 2 hours experiments. The quality of the crystals resulting from the shorter experiments (2 hours and 12 hours) was nevertheless considerably lower than from the experiments performed over a longer time range.

Complementary to this crystallization experiments with cyclodextrin, we tried to improve the quality of already formed crystals by adding phospholipase A2 to the the samples as proposed in [31]. In brief, phospholipase degrades the phospholipids and allows thereafter a denser protein packing in the crystal. The degradation products of the lipids would re-solve the existing crystals and need therefore to be removed from the solution or to be complexed. MBCD was used to complex the degradation products and thus to disable their solution potential. Figure 2.3 shows a schematic representation of these processes. The details of the processes and the results are described in [46].

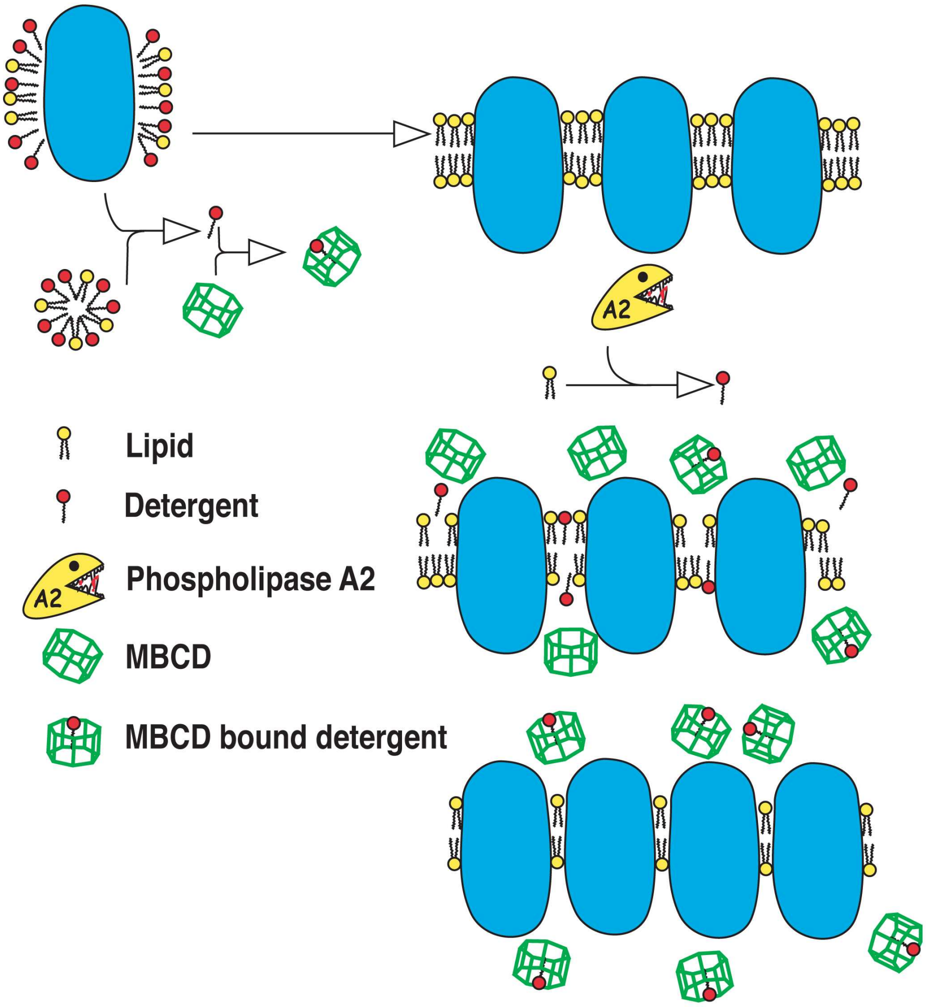


Figure 2.3: Detergent Removal
This scheme shows the principles of detergent removal, crystal formation and crystal quality improvement with phospholipase A2. After the initial detergent removal from the ternary mixture (top left), lipid double layers form and start to crystallize (top right). Lipid degradation using phospholipase A2 allows a denser packing of the integrated protein and improves thus the crystal quality (bottom). A re-solubilization of the crystal is avoided by complexing the lipid degradation products with MBCD.

2.2.2 Contributions

My contribution consisted of planing and setting up the cyclodextrin experiments with the help of Hervé Remigy. I collected the samples, prepared the grids and screened them at the electron microscope after initial supervision of Hervé Remigy. Acquisition of electron diffraction patterns of SoPIP2:1 was performed together with Wanda Kukulski who also provided the protein. Thomas Kaufmann performed the measurements for the cyclodextrin-detergent titration curves.

2.2.3 Cyclodextrin Paper

The results of this work were published 2006 in the Journal of Structural Biology under the title “Controlled 2D crystallization of membrane proteins using methyl-beta-cyclodextrin”. Following is the original paper, as published.

Available online at www.sciencedirect.com
ScienceDirect
Journal of Structural Biology 157 (2007) 321–328
Journal of Structural Biology
www.elsevier.com/locate/yjsbi

Controlled 2D crystallization of membrane proteins using methyl-β-cyclodextrin

Gian A. Signorell, Thomas C. Kaufmann, Wanda Kukulski, Andreas Engel, Hervé-W. Rémigy *

M. E. Müller Institute for Microscopy at the Biozentrum, University of Basel, Basel, Switzerland

Received 9 May 2006; received in revised form 21 July 2006; accepted 22 July 2006
Available online 2 August 2006

Abstract

High-resolution structural data of membrane proteins can be obtained by studying 2D crystals by electron crystallography. Finding the right conditions to produce these crystals is one of the major bottlenecks encountered in 2D crystallography. Many reviews address 2D crystallization techniques in attempts to provide guidelines for crystallographers. Several techniques including new approaches to remove detergent like the biobeads technique and the development of dedicated devices have been described (dialysis and dilution machines). In addition, 2D crystallization at interfaces has been studied, the most prominent method being the 2D crystallization at the lipid monolayer. A new approach based on detergent complexation by cyclodextrins is presented in this paper. To prove the ability of cyclodextrins to remove detergent from ternary mixtures (lipid, detergent and protein) in order to get 2D crystals, this method has been tested with OmpF, a typical β-barrel protein, and with SoPIP2;1, a typical α-helical protein. Experiments over different time ranges were performed to analyze the kinetic effects of detergent removal with cyclodextrins on the formation of 2D crystals. The quality of the produced crystals was assessed with negative stain electron microscopy, cryo-electron microscopy and diffraction. Both proteins yielded crystals comparable in quality to previous crystallization reports.

Keywords: 2D Crystallization; Membrane protein; Cyclodextrin; Electron microscopy

1. Introduction

Detergents are used to extract proteins from membranes and during subsequent purification to obtain samples suitable for crystallization. During this process the membrane protein structure and function are affected by the loss of interaction with the lipids. To reconstitute membrane proteins into a lipid bilayer in order to obtain 2D crystals, the detergent must be removed from a protein–lipid–detergent ternary mixture. There are mainly three ways to remove detergent: dialysis, adsorption to biobeads and dilution (Rigaud et al., 1997; Hasler et al., 1998; Remigy et al., 2003). However, all of these methods have their limitations. For dialysis, the rate of detergent removal is closely related to the critical micellar concentration (CMC) of the detergent (Jap et al., 1992; Kuhlbrandt, 1992). For example, dialysis of low CMC detergents (such as Triton X-100 or dodecyl-β-D-maltoside (DDM)) takes weeks at room temperature. Low CMC detergents are often necessary to keep the protein in its active form, as they are better substitutes for the bilayer than high CMC detergents. When low CMC detergents are required, the use of biobeads offers the advantage of a fast detergent removal. However, this can be a problem, since the high efficiency of detergent adsorption to biobeads may result in too fast detergent removal leading to aggregation primarily when small (<20 μl) reconstitution volumes are used. Therefore a minimal initial volume of ternary mixture of 100 μl is required for removing the detergent in a controlled manner. This will lead to compromises

---

* Corresponding author. Fax: +41 61 267 21 09.
E-mail address: herve.remigy@unibas.ch (H.-W. Rémigy).

1047-8477/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jsb.2006.07.011

in the selection of screening conditions, considering the limited amount of protein sample usually available. In a dilution experiment, the protein concentration should remain above 0.2–0.5 mg/ml and the initial detergent concentration should be as low as possible (Remigy et al., 2003). An advantage inherent to dilution is the possibility to slow down or even interrupt the reconstitution process at any time, thereby allowing for slow and controlled passage through the critical phase of crystal assembly. The major drawback of the dilution approach, however, is the inability to remove the detergent completely.

Here, we present an alternative approach to produce 2D crystals by detergent removal based on inclusion complexes with cyclodextrins. α-, β- or γ-cyclodextrins are ring shaped molecules made of 6, 7 and 8 glucose molecules, respectively. The non-polar environment inside the ring enables cyclodextrin to enclose hydrophobic or amphiphilic molecules like cholesterol or detergents. This technique was already used to perform reconstitution of active membrane proteins into membranes at a high lipid-to-protein ratio (LPR) (Degrip et al., 1998; Turk et al., 2000; Zampighi et al., 2003). The reconstitution rate is directly related to the amount of cyclodextrin added. The higher affinity of the inclusion compounds of cyclodextrin for detergents than for lipids prevents the LPR to change during reconstitution. Affinity tests between cyclodextrin and many anionic, nonionic and zwitterionic detergents of various CMC have been made at high LPR and have shown an almost complete lipid recovery (Degrip et al., 1998). A suitable cyclodextrin (α-, β- or γ-cyclodextrin) with a sufficiently high binding affinity can be found for most detergents (Degrip et al., 1998). The affinity of a detergent molecule for a cyclodextrin is largely determined by the fit of the detergent's hydrophobic moiety with the cyclodextrin cavity (Degrip et al., 1998). All alkyl-chain-containing detergents have high affinity with β-cyclodextrin. γ-Cyclodextrin should be considered when a detergent with a bulky hydrophobic chain is used. Fully functional reconstitution of membrane proteins with any kind of detergent is therefore possible.

The cyclodextrin approach needs to be adapted to produce proteoliposomes used for membrane protein crystallization. Finding the conditions promoting intermolecular and intramolecular interactions is mandatory, e.g., lipid–protein interactions need to be optimized by screening lipids of different nature and by varying the LPR. Since specific protein–protein interactions depend on the pH and the presence of specific counterions, the search space for optimal crystallization conditions is of an even higher dimensionality.

OmpF and SoPIP2;1 were used to test this new 2D crystallization approach. OmpF is a β-barrel membrane protein from *Escherichia coli* and SoPIP2;1 an α-helical aquaporin from spinach leaf plasma membrane, over-expressed in *Pichia pastoris* (Karlsson et al., 2003). Both proteins have their structures already determined by X-ray crystallography (Cowan et al., 1992; Tornroth-Horsefield et al., 2005). In the present work, methyl-β-cyclodextrin (MBCD) was selected for its high solubility and its high affinity for a wide range of detergents commonly used in membrane protein chemistry.

2. Materials and methods

2.1. *MBCD/detergent titration curve*

The detergents octyl-β,D-glucoside (OG) and dodecyl-*N*,*N*-dimethylamine-*N*-oxide (LDAO) were purchased from Anatrace (Ohio, USA), whereas octyl-polyoxyethylene (Octyl-POE) was purchased from Alexis (Lausen, Switzerland). All detergents were of high purity grade ($\geqslant 98\%$) and were used without further purification. MBCD–detergent mixtures were obtained by dilution of appropriately combined stock solutions with reagent-grade water produced by a Milli-Q filtration system ($>18\,\mathrm{M\Omega}$). The pipetted volumes were weighed on a balance (Mettler AE50) purchased from Mettler-Toledo (Greifensee, Switzerland). The different molar ratios were obtained by varying the MBCD concentration and keeping the detergent concentration constant above the corresponding CMC in order to have a constant surface tension reduction contribution from the detergent. Contact angles between a sample droplet and the supporting Parafilm were measured using a homemade device as described by Kaufmann et al. (2006). Measurements were carried out at room temperature.

For calculation purposes the mean molecular mass of the MBCD was determined by MALDI-TOF on a Bruker Scout 26 Reflex III instrument (Bruker Daltonik, Bremen, Germany). The mass spectrometric analysis revealed an average substitution rate of 1.77 methyl groups per glucose molecule, leading to a mean molecular mass of 1310.4 Da per MBCD molecule.

The molecular weight of Octyl-POE was calculated as the mean of the masses of all represented ethoxylated (EO) species ($n_{\mathrm{EO}} = 1$–11) which gives 372.52 g/mol for the hypothetical number of ethylene-oxide units $n_{\mathrm{EO}} = 5.5$. As density of Octyl-POE the value of 1.015 g/cm^3, determined by Rosenbusch et al. (2001) was used.

2.2. *OmpF and SoPIP 2;1 purification and reconstitution*

OmpF was produced in *E. coli*, purified as reported previously (Holzenburg et al., 1989) and solubilized in 1% Octyl-POE. SoPIP2;1, previously referred as PM28A, expressed in *Pichia pastoris* was purified according to Karlsson et al. (2003). The phospholipids used to prepare proteoliposomes were dissolved in chloroform at a concentration of 10 mg/ml, dried under a stream of argon, further dried overnight in a dessicator and weighed. Reagent-grade water was added and the lipid solutions were then sonicated for 2 min in a cold water bath using a Branson sonifier 250 tip at 20% of the maximum output power. An aqueous solution of the same detergent used for protein solubilization was added to the lipids to get a final detergent concentration of 1% (5 mg/ml dimyristoyl-phosphatidylcholine

(DMPC) lipids were solubilized in 1% Octyl-POE for OmpF, 5 mg/ml *E. coli* polar extract lipids in 1% OG for SoPIP2;1).

To have exactly defined starting conditions, the protein–detergent mixture was dialyzed overnight against the crystallization buffer containing detergent. OmpF was dialyzed at 4 °C overnight against 20 mM Hepes, pH 7, 100 mM NaCl, 10 mM $MgCl_2$, 3 mM NaN_3 and 1% Octyl-POE. SoPIP2;1 was dialyzed at 4 °C overnight against 20 mM Hepes, pH 7.5, 100 mM NaCl, 50 mM $MgCl_2$, 2 mM dithiothreitol, 3 mM NaN_3 and 1% OG. After dialysis the detergent concentration was measured as described above (Kaufmann et al., 2006) and adjusted to 1% Octyl-POE for OmpF and 1% OG for SoPIP2;1 by dilution with the corresponding buffer. The final protein concentrations before lipid addition were 1.5 and 1.2 mg/ml for OmpF and SoPIP2;1, respectively. DMPC was added to OmpF to cover LPRs ranging from 0.15 to 1 *E. coli* polar lipid extracts were added to SoPIP2; 1 to cover LPRs ranging from 0.3 to 0.4. After the detergent was removed by controlled MBCD addition (see below) at room temperature the samples were washed to eliminate the MBCD from the solution: after collection they were centrifuged for 15 min at 11 000 rpm using an Eppendorf centrifuge 5415C. The supernatant was discarded and the pellet resuspended in 400 μl detergent-free crystallization buffer. This procedure was repeated twice and after a final centrifugation the pellet was collected for cryo-electron microscopy (EM) characterization.

2.3. Controlled MBCD addition

MBCD was from Fluka. A dilution device described by Remigy et al. (2003) was used to perform the controlled addition of the MBCD. The MBCD solutions were obtained by addition of cyclodextrin to reagent grade water. The MBCD solution is supplied to a cuvette by a microsyringe. A valve prevents diffusion between the syringe and the cuvette. To keep the sample homogeneous, a stirrer is placed at the bottom of the cuvette. Fast reconstitutions lasting 30 min, 2 h and 12 h with initial ternary mixture volumes of 20 μl were performed. During these experiments the samples were constantly stirred. The volume of MBCD solution added was evaluated to compensate evaporation, and the concentration of MBCD in solution was calculated to exceed the minimal amount of cyclodextrin required (see MBCD/detergent titration curves) to remove completely the detergent from the ternary mixture (Kaufmann et al., 2006). For 30 min and 2 h, 10 μl of a 30% MBCD solution were added to the solution. For 12 h, 25 μl of a 10% MBCD solution were added. 30, 30 and 15 μl of crystalline sample were recovered from the 30 min, 2 h and 12 h experiments, respectively. Seventy-two hours and 144 h experiments were carried out with and without stirring with an initial ternary mixture volume of 40 μl. To overcome liquid loss during these longer experiments, bigger volumes of less concentrated cyclodextrin solutions were added. Four-hundred microliters of a 2% MBCD solution were added over 72 h and 600 μl of a 1.3% MBCD solution were added over 144 h. Two-hundred to 230 μl of crystalline sample were collected after the experiments.

2.4. Phospholipase A2 treatment

OmpF crystals were treated with bee venom phospholipase A2 (available from Sigma) to remove lipids from the proteoliposome and thus get closer crystal packing. This procedure was described by Mannella (1984). Loosely packed OmpF crystals (1 mg/ml) in a buffer containing 20 mM Hepes, pH 7, 100 mM NaCl, 10 mM $MgCl_2$, 3 mM NaN_3 and 1% Octyl-POE were used to perform this treatment. Ten microliters of OmpF were added to 2 ml of low salt buffer (0.005% MBCD, 1 mM Tris–HCl, 0.1 mM ethylene diamine tetra acetic acid, pH 7.5). 0, 0.3, 0.6 and 1.2 U/ml of phospholipase A2 were added to OmpF aliquoted in low salt buffer (1 mg of phospholipase corresponds to 1500 U). The mixture was incubated overnight at 4 °C. The solution was centrifuged in a Beckmann TL-100 ultracentrifuge for 90 min at 55 000 rpm (100 000*g*) at 4 °C. The supernatant was discarded and the pellet was resuspended in the low salt buffer.

2.5. Electron microscopy

Specimens for negative stain electron microscopy were prepared by adsorbing proteoliposomes onto carbon films previously rendered hydrophilic by glow discharging in air. The grids were washed three times with distilled water and stained with saturated uranyl acetate (2%). Micrographs were taken on a Hitachi H-7000 transmission electron microscope operated at 100 kV using magnifications of 5000–50 000. An optical bench with a laser beam was used to evaluate the diffraction quality of the micrographs. Well-ordered crystals were embedded in 2% glucose on molybdenum grids covered with a carbon film that was previously evaporated onto mica and floated on the grid. Electron diffraction patterns were recorded at low electron doses (<5 electrons per $Å^2$) on a Gatan 2K × 2K CCD camera with a Philips CM-200 FEG microscope operated at 200 kV.

3. Results

3.1. MBCD/detergent titration

The titration curves (Fig. 1) reveal the amount of MBCD needed for the neutralization of the respective detergents. All three curves show a sudden increase in the contact angle reflecting the MBCD–detergent complexation depleting the solution from detergent monomers, which can no more adsorb to the liquid–vapor and solid–liquid interfaces. The points at which the curves reach the plateau correspond to the molar ratios needed for complete complexation of the detergents by MBCD, i.e., the minimal amount required in an experiment. At the same time these

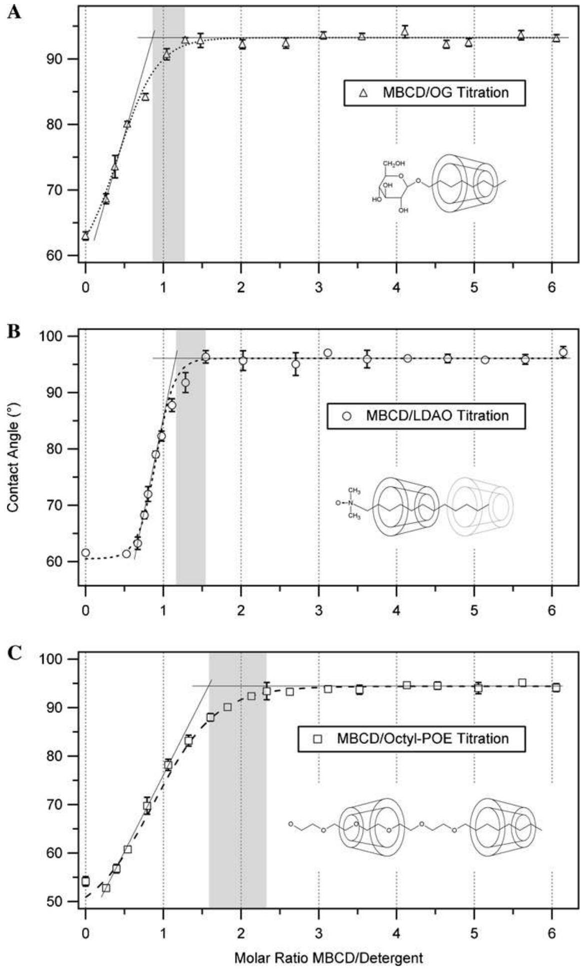


Fig. 1. MBCD–detergent titration curves. The measurements were performed at constant detergent concentration above the respective CMC of the detergents. One curve represents the mean of three measurements for each MBCD–detergent pair. The possible ranges for the stoichiometry of complex formation are shaded in gray. The illustrations depict the most probable inclusion stoichiometries as deduced from the measurements. In the case of the polydisperse Octyl-POE, pentaethylene glycol monooctyl ether (C_8E_5) is shown as a representative structure. (A) OG at 21 mM (CMC = 25 mM). (B) LDAO at 2.5 mM (CMC = 1.9 mM). (C) Octyl-POE at 10 mM (CMC = 6.6 mM).

points correspond to the upper boundary of the complex stoichiometry. The lower boundary is given by the intersection between the extrapolated plateau and the extrapolated initial slope reflecting the highest possible association constant for the complex formation.

The obtained ranges for the stoichiometries of complex formation are 0.9–1.3 for MBCD/OG, 1.2–1.5 for MBCD/LDAO and 1.6–2.3 for MBCD/Octyl-POE. The results suggest a 1:1 complex for MBCD and OG. In the case of LDAO the results comply with the formation of a predominant 1:1 complex and with a fraction of the species carrying a second MBCD molecule but with much lower affinity. The mean complex stoichiometry for the polydisperse Octyl-POE levels off at a molar ratio of around 2. This suggests that while the aliphatic chain is occupied by one MBCD molecule, the polyoxyethylene chain, which is variable in length, can be decorated with different amounts of MBCD molecules.

3.2. 2D Crystallization of the porin OmpF

Experiments ranging from 30 min to 144 h yielding OmpF crystals show that faster detergent removal rates (2 h and less) result in low quality crystals having diameters ranging from 50 to 500 nm, with a trigonal lattice ($a = b = 9 \pm 0.5$ nm; $\gamma = 60°$), which diffracted up to 2.7 nm resolution (Fig. 2A). Slower rates (12–144 h) result in the formation of large sheets having diameters ranging from 1 to 2 μm, with a trigonal lattice ($a = b = 9 \pm 0.5$ nm; $\gamma = 60°$) diffracting up to 2.4 nm resolution (Fig. 2B). Stirring did not affect the crystal quality. After phospholipase A2 treatment (Mannella, 1984) on OmpF crystals obtained after 144 h runs, crystals with a trigonal lattice ($a = b = 7.2 \pm 0.2$ nm; $\gamma = 60°$) diffracting beyond 2.2 nm resolution (Fig. 2C) were obtained.

3.3. 2D Crystallization of SoPIP2;1

SoPIP2;1 reconstitutions lasting 2 h (Fig. 3A) and less yielded large vesicles and small sheets but no significant diffraction was observed. Large sheets comparable to those previously reported (Kukulski et al., 2005) were obtained after 72 and 144 h experiments (Fig. 3B). Stirring did not affect the crystal quality. The largest sheets obtained after 144 h were used to perform direct electron diffraction. Images of electron diffraction patterns were taken from unwashed (Fig. 3C) and washed crystals (Fig. 3D). The electron diffraction patterns exhibit a p4 symmetry with the same lattice constants of $a = b = 9{,}6$ nm for both, unwashed and washed samples. The indicated spots for the unwashed crystals are 24,2 (Fig. 3C) and 24,3 for washed crystals (Fig. 3D), corresponding to a resolution of 4 Å. The unwashed sample (Fig. 3C) displays strong additional diffraction spots, arranged in a sixfold symmetry overlaying the typical diffraction pattern of SoPIP2;1. These spots are not present in the washed sample (Fig. 3D).

4. Discussion

The capability of cyclodextrin to complex any kind of detergent molecule, independently from the CMC, is a crucial advantage over the dialysis method. The removal of detergents like Triton TX-100 or DDM can be performed. Such “mild” detergents are widely used for the purification of large and sensitive complexes like *Chlorobium tepidum* reaction center (Remigy et al., 1999). The nature of the detergent, the detergent removal rate and the detergent

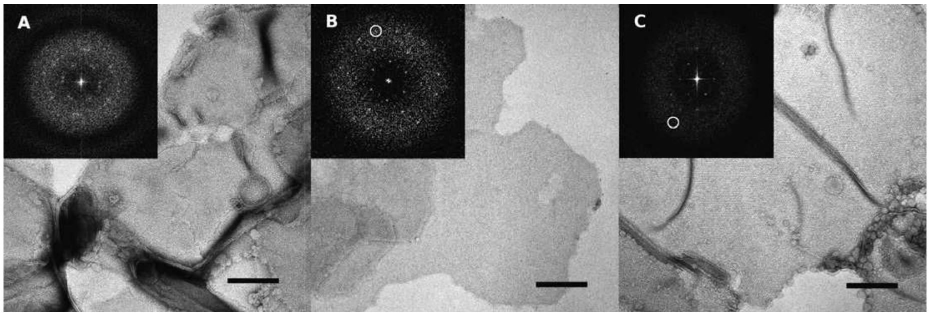


Fig. 2. OmpF 2D crystallization using the MBCD procedure. Experiments ranging from 30 min to 144 h show that (A) faster detergent removal rates (2 h and less) result in low quality crystals having diameters ranging from 50 to 500 nm, with a trigonal lattice ($a = b = 9 \pm 0.5$ nm; $\gamma = 60°$). (B) Slower rates (12–144 h) result in the formation of large sheets having a diameter ranging from 1 to 2 μm, with a trigonal lattice ($a = b = 9 \pm 0.5$ nm; $\gamma = 60°$) diffracting up to 2.4 nm resolution (spot marked by a circle). (C) After phospholipase A2 treatment on OmpF crystals obtained after a 144 h run, crystals with a trigonal lattice ($a = b = 7.2 \pm 0.2$ nm; $\gamma = 60°$) diffracting beyond 2.2 nm resolution were obtained (spot marked by a circle). The scale bars represent 100 nm.

removal technique affect size and quality of the resulting proteoliposomes and crystals (Wingfield et al., 1979; Hovmoller et al., 1983; Chami et al., 2001; Remigy et al., 2003). Even if the detergent is removed in an efficient way, there is no guarantee that 2D crystals will form during the reconstitution process. For this reasons the validation of the cyclodextrin approach in its ability to produce membrane protein crystals was needed.

4.1. Cyclodextrin and detergent removal

To perform reconstitution with MBCD accurately, a precise evaluation of the amount of MBCD needed to remove all the detergent from a solution is required. A homemade device (Kaufmann et al., 2006) to measure the detergent concentration of any solution (detergent solution, binary or ternary mixtures) was used to measure the MBCD–detergent molecular ratios after cyclodextrin addition to detergent solutions. The insets in Fig. 1 show the most probable complexes between MBCD and the detergents. The obtained ratios for OG (1:1), LDAO (1:1 and a weak 2:1) and Octyl-POE (2:1) are in good agreement with expected values based on the alkyl-chain length. The cavity of a β-cyclodextrin molecule is about 8 Å deep, offering accommodation for a C8 chain (Fig. 1A). The longer chain of LDAO might be shielded by an additional cyclodextrin ring occasionally forming a 2:1 complex (Fig. 1B). Alternatively a 3:2 complex could also be envisaged, where two LDAO molecules carry one MBCD each and an additional MBCD molecule is shared between the two. This, however, is purely speculative and might not be very probable for thermodynamic reasons. Octyl-POE has a lower stoichiometry than expected. One MBCD molecule is strung onto the C8 aliphatic chain, whereas only one more is occupying the polyethylene oxide (PEO) chain (Fig. 1C). The latter could reflect a lower affinity for PEO fragments due to the oxygen atoms possibly hydrogen bonding with surrounding water molecules. Thus, the change of free energy is lower for PEO units. However, the polydisperse character of Octyl-POE clearly interferes with the assignment of a single stoichiometric ratio and this might be reflected in the comparatively smaller gradient of the curve. Additionally, one should keep in mind that in solution both the aliphatic chains and the polyoxyethylene chains are not only present as an extended species and therefore the geometric reasoning is of approximative nature.

Schmidt-Krey et al. (1998) have shown that the size of microsomal glutathione transferase 2D crystals changes according to the initial detergent concentration. The MBCD approach allows us to use an excess of detergent even with low CMC detergents. Having established MBCD–detergent titration curves (Figs. 1A–C) we can calculate the exact amount of MBCD that needs to be added to a solution to remove all the detergent. For practical purposes one needs to add the highest molar ratio (see Section 3.1) since complete neutralization of the detergent is required. Also, since the detergent removal rate is directly related to the cyclodextrin addition, a mixture of detergents can be reproducibly used during crystallization. Hence, one can easily combine low and high CMC detergents without the drawbacks related to dialysis. This enables us to explore 2D crystallization conditions that could not be tested yet.

4.2. 2D Crystallization of OmpF

Using the cyclodextrin approach to remove detergent produced OmpF crystals of different qualities depending on the reconstitution time. Short reconstitution periods (up to 2 h) yielded low quality crystals (Fig. 2A) similar to the crystals reported by Dorset et al. (1983). Longer reconstitution experiments (12–144 h) led to bigger proteoliposomes having an average diameter of 1.5 μm that diffracted up to 2.4 nm (Fig. 2B). These results are in agreement (according to the size of the proteoliposomes and the lattice parameters) with previous works on OmpF (Dolder et al., 1996; Remigy et al., 2003) where dilution and dialysis yielded double-layered trigonal crystals. To improve the crystal packing of OmpF, we adapted the procedure using phospholipase A2 (PLA2) described by

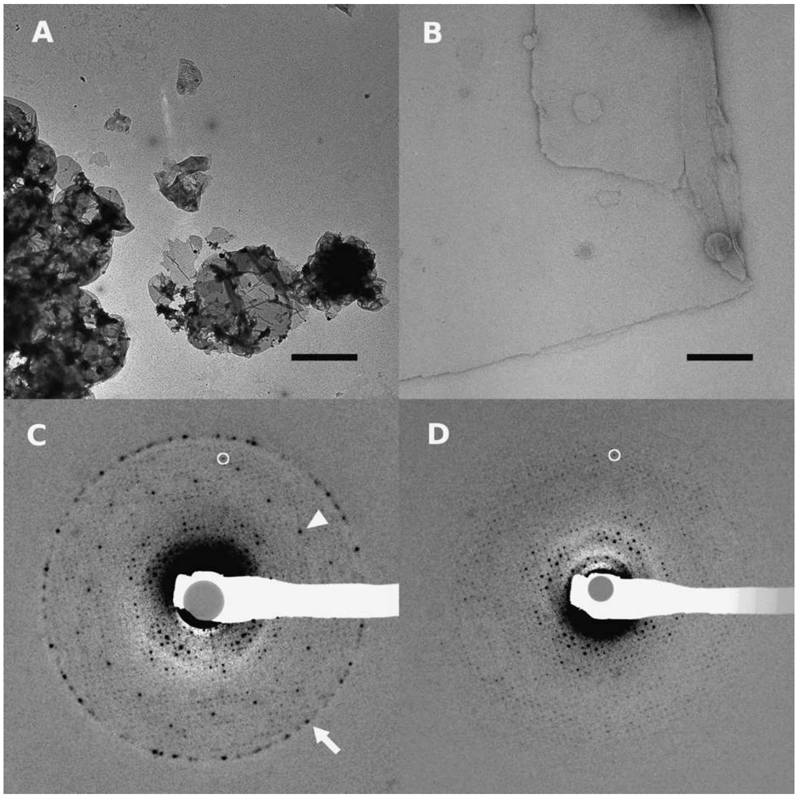


Fig. 3. SoPIP2;1 2D crystallization using MBCD. (A and B) Electron microscopy images using negative stain. (A) Overview at 5k magnification of a 2 h run and (B) sheet at 50k magnification of a crystal obtained after a 144 h reconstitution. The scale bar in (A) represents 1 μm and in (B) 100 nm. (C and D) Electron diffraction patterns of crystals obtained after 144 h. Both images show a p4 symmetry with lattice constants of $a = b = 96$ Å. (C) The unwashed sample shows strong additional diffraction spots arranged in a sixfold symmetry (arrowhead). The spots arranged in circle are from water crystals (arrow). The spot marked by a circle on the diffraction image is indexed 24,2. (D) The washed sample shows no additional spot overlaying the typical diffraction pattern of SoPIP2;1. The spot marked by a circle (24,3) corresponds to a resolution of 4 Å.

Mannella (1984) to the cyclodextrin method. PLA2 catalyzes the hydrolysis of the ester linkage in the sn-2 position of glycerophospholipids like DMPC, yielding free fatty acids and lysophospholipids. To avoid the solubilization of the proteoliposomes by these degradation products the solution is dialyzed against a low salt buffer (Mannella, 1984). The ability of cyclodextrin to bind free fatty acids and lysophospholipids allowed us to perform the PLA2 treatment without dialysis and thus to simplify the procedure. After PLA2 treatment the packing of the proteins was denser and a smaller trigonal lattice was revealed (Fig. 2C). These crystals are similar to the densely packed OmpF crystals that have undergone a PLA2 treatment described by Regenass et al. (1985) and Engel et al. (1992). Due to the limitations of negative staining these crystals diffracted only up to 2.2 nm (Fig. 2C). Thus PLA2 treatment can be applied directly on loosely packed crystals obtained with the MBCD method given that there is enough MBCD to capture the digested lipids. This approach is appropriate when the initial amount of lipid is higher than is optimal for crystallization. This can be required to keep the protein stable during the reconstitution step.

4.3. 2D Crystallization of SoPIP2;1

Reconstitutions of SoPIP2;1 lasting 2 h and less did not yield large sheets and did not diffract in negative stain (Fig. 3A). The weak contrast in negative stained samples is due to the smooth surface of double-layered SoPIP2;1 crystals preventing negative stain to penetrate (Kukulski, personal communication). Only cryo-EM could correctly assess the quality of such crystals. Reconstitutions of SoPIP2;1 over 144 h produced large crystals (Fig. 3B) so that electron diffraction could be performed. The diffraction patterns (Figs. 3C and D) correspond to one type of 2D crystal, obtained by dialysis during the determination of the 5 Å structure of SoPIP2;1 (Kukulski et al., 2005). The quality of the crystals in the present work is comparable, having spots diffracting up to a resolution of 4 Å. The unwashed sample (Fig. 3C) displays strong additional diffraction spots arranged in a sixfold symmetry and overlaying the typical diffraction pattern of SoPIP2;1. These spots disappear when the sample is washed (Fig. 3D). Because washing mainly removed the MBCD from the solution, we assume that the spots in the unwashed sample result from MBCD crystallization during grid preparation.

4.4. Large screenings for 2D crystals using cyclodextrin

To parallelize 2D crystallization experiments to test as many conditions as possible, it is important to have the smallest possible initial volume of ternary mixtures. The accuracy in the addition of salts, lipids and other compounds imposes a limit in reducing this initial sample volume. Evaporation poses the problem of keeping the sample volume constant over long time spans, which is a difficult task to handle, especially with small volumes. Using microdroplet pipettes to dispense cyclodextrin solution and water in combination with an accurate volume measurement method could overcome these drawbacks. We estimate that small volumes in the order of 10 µl can be handled automatically. In this way up to 100 different reconstitution conditions could be tested with 1 mg of protein. This is a decisive step towards the reproducibility of 2D crystallization experiments. Still remaining is the problem of preparing and inspecting a large quantity of individual specimens by electron microscopy (Kuhlbrandt, 1992).

In our experiments crystallization mixtures were dialyzed overnight to have precise starting conditions. To diversify conditions (salts concentrations, LPRs etc.) it makes sense to dialyze batches of detergent–protein mixtures against buffers of various pHs overnight and to add subsequently additional substances to the solution (lipids, salts, inhibitors etc.).

Short experiments (12 h and less) yielded crystals of lower quality (small and less ordered), whereas large sheets were obtained after 72 h and more for both OmpF and SoPIP2;1. From these results we assume that longer duration will in general increase the size and the quality of the crystals. Also, over longer experiments (72 and 144 h) mixing did not affect crystal quality. To achieve homogeneous conditions over longer reconstitutions, a stirrer was used in our setup. In parallel experiments a shaker would be more appropriate, since shaking does not interfere with the sample directly (no contamination) and commercially available microplates (with 96 or 384 wells) could be used. A more sophisticated device than the dilution apparatus, controlling all pertinent parameters of the ternary mixture during crystallization would enable us to perform longer experiments promoting crystallogenesis in a reproducible manner.

The transition phase temperature increases according to the ratio of saturated/unsaturated fatty acid chains of lipids. Therefore, the temperature during reconstitution is a crucial parameter (Engel et al., 1992). For example, the phase transition temperature of DMPC, commonly used for crystallization, is beyond 23 °C (Nakayama et al., 1980). The effects of the temperature as parameter on the crystallization using cyclodextrins have not been investigated extensively. Since MBCD is highly soluble in water even at low temperatures, the described procedure shows no restriction in experimenting any temperature profiles during crystallization.

The reconstitution and crystallization using the cyclodextrin approach requires only small sample volumes and no additional surrounding buffer like in the dialysis method. This makes the cyclodextrin method very suitable to screen conditions where expensive or difficult to produce compounds are used (proteins, chemicals, substrates or inhibitors). A certain protein conformation or better-ordered crystals may be achieved if such compounds (ATP, antagonists or substrates) are added to the ternary mixture. In the dialysis method these compounds have to be added to the dialysis buffer solution to keep their concentrations constant since they are usually small enough to pass through the dialysis membrane. Many examples of co-crystallization in the 3D crystallization field involving inhibitors with high affinity, transiently bound substrates (e.g., NAD^+), designed protein (e.g., ankyrin repeats), or other cofactors have been reported (Schindler et al., 2000; Scott et al., 2004; Brautigam et al., 2005; Kohl et al., 2005; Sundaresan et al., 2005). Such conditions have not being explored by 2D crystallization because of the limitations mentioned above. Therefore, the cyclodextrin approach appears to be a promising alternative to traditional 2D crystallization methods.

5. Conclusion

Cyclodextrin can be used in protein reconstitution, crystallization and to improve crystal quality in combination with phospholipase. Proteins of known structure were chosen to test this new method of 2D crystallography. The quality of both OmpF and SoPIP2;1 crystals were comparable with previously published results (Dolder et al., 1996; Remigy et al., 2003; Kukulski et al., 2005). One advantage of this method is the accuracy of the detergent removal, allowing us to control the kinetics of the whole process in a precise way. The detergent removal rate is controlled by the amount of cyclodextrin added and therefore does not depend on the CMC of the detergent. Another advantage of the cyclodextrin method over other methods lies in its applicability in systematic screenings for crystallization conditions. The sample volume can be very small allowing to work with a small amount of protein per condition and with compounds that are expensive or not available in large quantities. The possibility of large parallel screenings of 2D crystallization conditions needs to be complemented with the effort to automate electron microscopy in such a way that all the conditions can be inspected in a reasonable time. The approach needs to be explored with a large range of membrane proteins in order to acquire a solid know-how of kinetics, choice of detergents and choice of the right cyclodextrin partner. Only these systematic experiments will give us the knowledge to produce 2D crystals of any membrane protein in a reproducible way.

Acknowledgments

We are grateful to Per Kjellbom for providing *Pichia pastoris* over-expressing SoPIP2;1 and Mohamed Chami for fruitful feedback on membrane protein reconstitution. Heiko Heerklotz is acknowledged for critically discussing the results of detergent complexation by MBCD. This work was supported by the NCCR Nano, the NCCR Structural Biology, the SNF grant to Andreas Engel (SNF 501 221) and the NoE 3DEM.

References

Brautigam, C.A., Chuang, J.L., Tomchick, D.R., Machius, M., Chuang, D.T., 2005. Crystal structure of human dihydrolipoamide dehydrogenase: NAD$^+$/NADH binding and the structural basis of disease-causing mutations. J. Mol. Biol. 350, 543–552.

Chami, M., Pehau-Arnaudet, G., Lambert, O., Ranck, J.L., Levy, D., Rigaud, J.L., 2001. Use of octyl-β-thioglucopyranoside in two-dimensional crystallization of membrane proteins. J. Struct. Biol. 133, 64–74.

Cowan, S.W., Schirmer, T., Rummel, G., Steiert, M., Ghosh, R., Pauptit, R.A., Jansonius, J.N., Rosenbusch, J.P., 1992. Crystal structures explain functional properties of two *E. Coli* porins. Nature 358, 727–733.

Degrip, W.J., Vanoostrum, J., Bovee-Geurts, P.H., 1998. Selective detergent-extraction from mixed detergent/lipid/protein micelles, using cyclodextrin inclusion compounds: a novel generic approach for the preparation of proteoliposomes. Biochem. J. 330, 667–674.

Dolder, M., Engel, A., Zulauf, M., 1996. The micelle to vesicle transition of lipids and detergents in the presence of a membrane protein: towards a rationale for 2D crystallization. FEBS Lett. 382, 203–208.

Dorset, D.L., Engel, A., Haner, M., Massalski, A., Rosenbusch, J.P., 1983. Two-dimensional crystal packing of matrix porin. A channel forming protein in *Escherichia coli* outer membranes. J. Mol. Biol. 165, 701–710.

Engel, A., Hoenger, A., Hefti, A., Henn, C., Ford, R.C., Kistler, J., Zulauf, M., 1992. Assembly of 2-D membrane protein crystals: dynamics, crystal order, and fidelity of structure analysis by electron microscopy. J. Struct. Biol. 109, 219–234.

Hasler, L., Heymann, J.B., Engel, A., Kistler, J., Walz, T., 1998. 2D crystallization of membrane proteins: rationales and examples. J. Struct. Biol. 121, 162–171.

Holzenburg, A., Engel, A., Kessler, R., Manz, H.J., Lustig, A., Aebi, U., 1989. Rapid isolation of OmpF porin-LPS complexes suitable for structure-function studies. Biochemistry 28, 4187–4193.

Hovmoller, S., Slaughter, M., Berriman, J., Karlsson, B., Weiss, H., Leonard, K., 1983. Structural studies of cytochrome reductase. Improved crystals of the enzyme complex and crystallization of a subcomplex. J. Mol. Biol. 165, 401–406.

Jap, B.K., Zulauf, M., Scheybani, T., Hefti, A., Baumeister, W., Aebi, W., Engel, A., 1992. 2D crystallization: from art to science. Ultramicroscopy 46, 45–84.

Karlsson, M., Fotiadis, D., Sjovall, S., Johansson, I., Hedfalk, K., Engel, A., Kjellbom, P., 2003. Reconstitution of water channel function of an aquaporin overexpressed and purified from Pichia pastoris. FEBS Lett. 537, 68–72.

Kaufmann, T.C., Engel, A., Remigy, H.W., 2006. A novel method for detergent concentration determination. Biophys. J. 90, 310–317.

Kohl, A., Amstutz, P., Parizek, P., Binz, H.K., Briand, C., Capitani, G., Forrer, P., Plückthun, A., Grütter, M., 2005. Allosteric inhibition of aminoglycoside phosphotransferase by a designed ankyrin repeat protein. Structure 13, 1131–1141.

Kuhlbrandt, W., 1992. Two-dimensional crystallization of membrane proteins. Q. Rev. Biophys. 25, 1–49.

Kukulski, W., Schenk, A.D., Johansson, U., Braun, T., de Groot, B.L., Fotiadis, D., Kjellbom, P., Engel, A., 2005. The 5Å structure of heterologously expressed plant aquaporin SoPIP2;1. J. Mol. Biol. 350, 611–616.

Mannella, C.A., 1984. Phospholipase-induced crystallization of channels in mitochondrial outer membranes. Science 224, 165–166.

Nakayama, A., Mitsui, T., Nishihara, M., Kito, M., 1980. Relation between growth temperature of *E. coli* and phase transition temperatures of its cytoplasmic and outer membranes. Biochim. Biophys. Acta 601, 1–10.

Regenass, M., Hardmeyer, A., Rosenbusch, J.P., Engel, A., 1985. Phospholipids in reconstituted porin membranes: conversion of one crystal habit in another by phospholipase. Experientia 41, 808.

Remigy, H.W., Stahlberg, H., Fotiadis, D., Müller, S.A., Wolpensinger, B., Engel, A., Hauska, G., Tsiotis, G., 1999. The reaction complex from the green sulfur bacterium *Chlorobium tepidum*: a structural analysis by scanning transmission electron microscopy. J. Mol. Biol. 290, 851–858.

Remigy, H.W., Caujolle-Bert, D., Suda, K., Schenk, A., Chami, M., Engel, A., 2003. Membrane protein reconstitution and crystallization by controlled dilution. FEBS Lett. 555, 160–169.

Rigaud, J.L., Mosser, G., Lacapere, J.J., Olofsson, A., Levy, D., Ranck, J.L., 1997. Bio-beads: an efficient strategy for two-dimensional crystallization of membrane proteins. J. Struct. Biol. 118, 226–235.

Rosenbusch, J.P., Lustig, A., Grabo, M., Zulauf, M., Regenass, M., 2001. Approaches to determining membrane protein structures to high resolution: do selections of subpopulations occur? Micron 32, 75–90.

Schindler, T., Bornmann, W., Pellicena, P., Miller, W.T., Clarkson, B., Kuriyan, J., 2000. Structural Mechanism for STI-571 inhibition of Abelson Tyrosine Kinase. Science 289, 1938–1942.

Schmidt-Krey, I., Lundqvist, G., Morgenstern, R., Hebert, H., 1998. Parameters for the two-dimensional crystallization of the membrane protein microsomal glutathione transferase. J. Struct. Biol. 123, 87–96.

Scott, E.E., White, M.A., He, Y.A., Johnson, E.F., Stout, C.D., Halpert, J.R., 2004. Structure of mammalian cytochrome P450 2B4 complexed with 4-(4-Chlorophenyl) imidazole at 1.9-Å resolution. J. Biol. Chem. 279, 27294–27301.

Sundaresan, V., Chartron, J., Yamaguchi, M., Stout, C.D., 2005. Conformational diversity in NAD(H) and interacting transhydrogenase nicotinamide nucleotide binding domains. J. Mol. Biol. 346, 617–629.

Tornroth-Horsefield, S., Wang, Y., Johanson, U., Karlsson, M., Tajkhoshid, D., Neutze, R., Kjellbom, P., 2005. Structural mechanism of plant aquaporin gating. Nature 439, 688–694.

Turk, E., Kim, O., le Coutre, J., Whitelegge, J.P., Eskandari, S., Lam, J.T., Kreman, M., Zampighi, G., Faull, K.F., Wright, E.M., 2000. Molecular characterization of *Vibrio parahaemolyticus* vSGLT: a model for sodium-coupled sugar cotransporters. J. Biol. Chem. 275, 25711–25716.

Wingfield, P., Arad, T., Leonard, K., Weiss, H., 1979. Membrane crystals of ubiquinone: cytochrome c reductase from *Neurospora* mitochondria. Nature 280, 696–697.

Zampighi, G.A., Kreman, M., Lanzavecchia, S., Turk, E., Eskandari, S., Zampighi, L., Wright, E.M., 2003. Structure of functional single AQP0 channels in phospholipid membranes. J. Mol. Biol. 325, 201–210.

2.2.4 Outlook

The method described in the paper proved to be a good alternative to the crystallization with dialysis buttons. The lower protein consumption, the easier handling and the shorter time requirements using low CMC detergents being the main advantages of this method.

The experiments described in the paper were performed using a device originally designed to generate 2D crystals by diluting the samples in order to remove the detergent [39]. In figure 2.4, an overview picture of the machine used to perform the experiments described in [46] can be seen. As the used setup was not originally designed to perform cyclodextrin experiments, some problems were encountered during the various experiments. The setup of the dilution machine represents an open system, exposing the samples to environment. The resulting evaporation leads to a non predictable decrease of volume during the experiment duration. Thus, the amount of added cyclodextrin solution had to be overestimated to prevent the sample from drying out. The mixing of the sample, which is done with a small screw-driver like device, inserted directly into the sample (see figure 2.5) reaches only a part of the sample and generates a certain inhomogeneity during the addition of cyclodextrin

Most of these problems could be overcome by a machine allowing accurate volume control in combination with a feedback controlling the amount of liquid being added at any time. A device addressing this issues is currently constructed by Hervé Remigy. It uses a 96 well-plate as sample container and a capacitive sensor to measure the liquid level in every well. According to the measured liquid volume, a certain amount of water will be added in addition to the cyclodextrin solution to compensate for the evaporation. The mixing issues are addressed in the new setup by shaking the whole well-plate, resulting in a more homogeneous mixing without contamination. The automated device will further allow to use even smaller sample volumes for the cyclodextrin experiments. An additional benefit of this setup will be the larger number of conditions that can be screened in parallel (up to 96 conditions as opposed to 6 to 8 when the dilution machine is used).

The automation of the cyclodextrin experiments will lead to other bottlenecks, namely in the sample preparation and in the sample screening at the microscope. The preparation of the initial sample conditions will be done automatically using a commercially available mixing robot. Such a robot is available in our lab and the setup is currently taking place. Concerning automated sample screening, there are still hard to solve problems, since the analysis of crystallization samples needs a good amount of human expertise and experience. The localization of crystals is hard to achieve without user interaction. Nevertheless a project has been set up within the framework of the HT3DEM projects that will address and hopefully solve these problems.

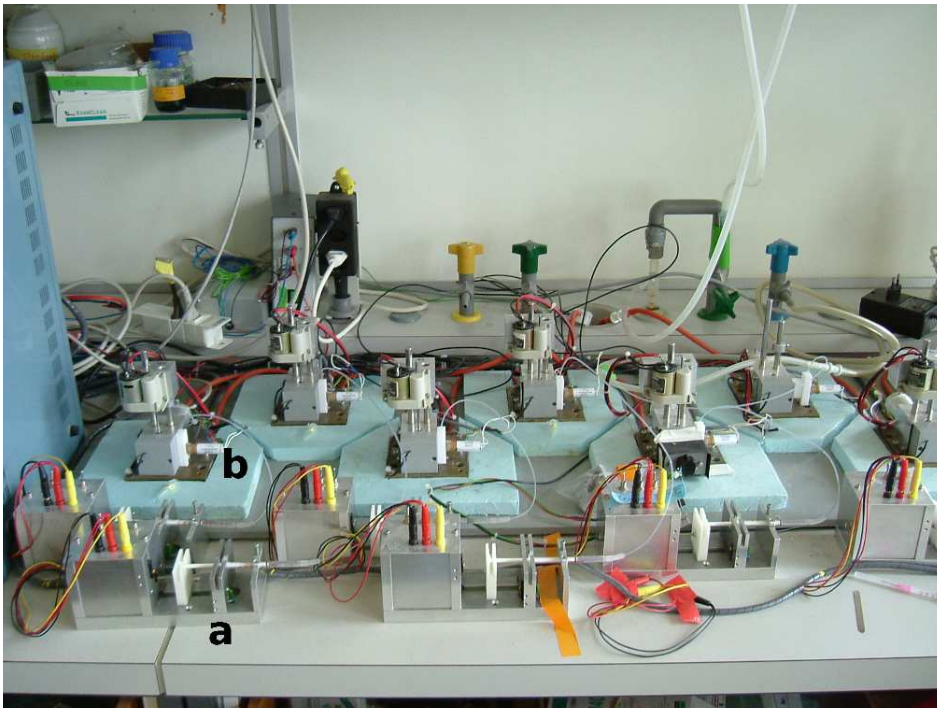


Figure 2.4: Overview of the Dilution Machine
The dilution machine as used for our experiments consists of eight independently operable entities. Each entity consists of a syringe for the cyclodextrin addition (a) controlled by a special software and a reaction chamber (b) containing the initial ternary mixture. The back-flow from the reaction chamber (b, see Fig. 2.5) is prevented by a valve which is only opened upon liquid addition.

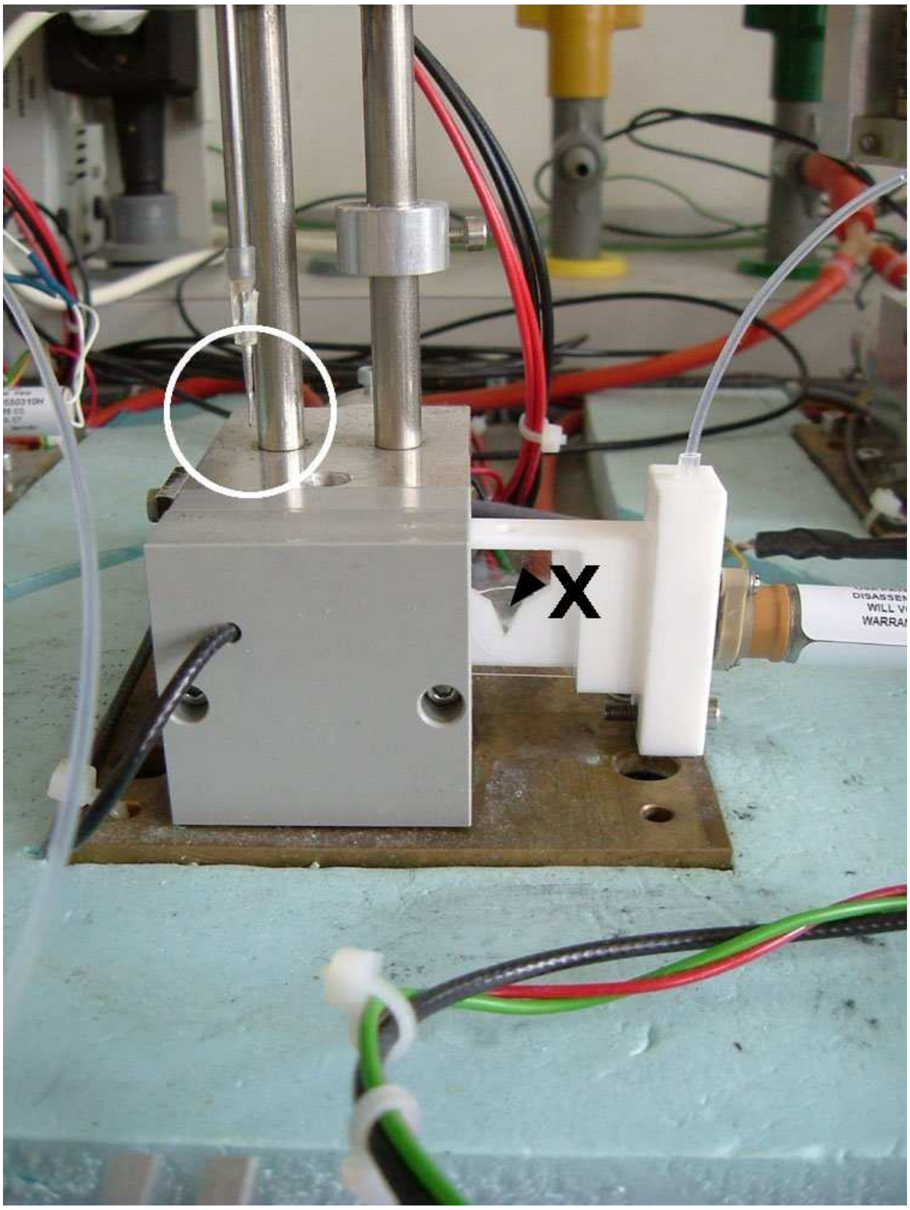


Figure 2.5: Reaction Chamber and Mixing Device
The mixing device of the dilution machine (white circle) inserts into the sample (X) and turns according to the applied voltage at the electromotor. The ternary mixture (sample) is added into the reaction chamber through the opening also used for the insertion of the mixing device. The cyclodextrin solution is added to the sample through a tube connected to the syringe.

2.2.5 Methyl-β-cyclodextrin (MBCD) Experiments Using other Proteins

In the development of the MBCD crystallization setup, we could experiment with a variety of proteins from different sources. The low number of machines, the sometimes small amount of protein and in some cases also the quality of the protein prevented us from investigating the crystallization conditions in a systematic way for most of these proteins. Prior to our experiments the crystallization conditions for all the tested proteins, except AQP1, were unknown. Some insight about crystallization conditions and about the setup of the experiments could be gained with these experiments. The experiments could be performed much faster than they could have been with dialysis buttons and less protein was used per condition. Experience on how much the evaporation influences the experiment, the time range of the experiments (kinetics of detergent removal by cyclodextrin) and the handling of the dilution machine used for our MBCD experiments could be gained from these trials.

In the following sub chapters a summary of the experiments and their outcome for every of the tested proteins is presented.

2.2.5.1 AQP1

In the early development of the cyclodextrin method for detergent removal, experiments with AQP1 were performed. As we had no prior experience about the behaviour of this method, different time ranges over which the detergent was removed were tested. Buffer composition, LPR and detergent were chosen according to the experience of AQP1 crystallization performed with the dialysis method as described in [53] and according to the experiences from other group members.

Materials and Methods AQP1 was expressed and purified as described in [53]. The protein concentration was determined to be 2.7 mg/ml with a BCA (bicinchoninic acid) assay from Pierce [47]. The protein was in 2% OG. The sample was diluted to a concentration of 2 mg/ml with a buffer containing 20 mM MES, pH 6, 200 mM NaCl, 2 mM DTT (dithiothreitol), 2mM $NaN_3$and 10 mM $MgCl_2$. To have exactly defined starting conditions, the protein-detergent mixture was dialyzed overnight at 4°C against 50 ml of the same buffer containing 1% OG. Protein and detergent concentration were measured and the detergent concentration was adjusted to 1% OG by dilution with the described buffer (without detergent). The final protein concentration before lipid addition was 1.5 mg/ml. *E.Coli* lipid (at 5 mg/ml in 3% OG) was added to reach an LPR of 1. This ternary mixture was incubated for 2h at room temperature.

DDM was used as detergent in the 16h and 44h experiments. The protein was solubilized in 1% DDM and had a concentration of 1.7 mg/ml (before detergent addition). *E. Coli* lipid (at 4 mg/ml in 4% DDM) was added to reach an LPR of 1. This ternary mixture was incubated for 2h at room temperature.

The detergent was removed using the method described in Signorell et al. [46]. The detergent removal was conducted over different time ranges: Reconstitutions lasting 30’, 2h, 12h, 16h, 44h and 72 h were performed. The amount of MBCD calculated to remove all the detergent from the ternary mixture was solubilized in a volume of reagent grade water (for the 16h and 44h experiments, the buffer described above was used to solubilize the cyclodextrin) to compensate the evaporation. The concentration of MBCD and as a function of that, the volume of the added liquid was dependent on the reconstitution time: 10 μl of a 30% MBCD solution was added over 30’ to 20 μl ternary mixture (recovered volume ~25 μl); 10μl of a 30% MBCD solution was added over 2h to 20 μl ternary mixture (recovered volume ~30 μl); 25μl of a 10% MBCD solution was added over 12h to 20 μl ternary mixture (recovered volume ~30 μl); 100μl of a 2% MBCD solution was added over 16h and 44h to 20 μl ternary mixture and 200μl of a 0.5% MBCD solution was added over 172h to 20 μl ternary mixture.

Specimens for negative stain electron microscopy were prepared by adsorbing proteoliposomes onto carbon films previously rendered hydrophilic by glow discharging. The grids were washed three times with distilled water and stained with 1% uranyl acetate. Micrographs were taken on a Hitachi H-7000 transmission electron microscope operated at 100 kV using magnifications of 5000 and 50’000.

Results and Discussion Most of the parlodion film on the EM-grids from the samples containing OG in the initial ternary mixture was destroyed. The parts of the film that were not broken showed a strong background and only small particles were visible. The destruction of the plastic grids lead to the assumption, that not all detergent was removed from the samples. The high concentrations of MBCD in the recovered samples and the still solubilized protein and lipid was responsible for the high background.

The full amount of cyclodextrin solution was added at once to the sample recovered after 12h to test if the remaining detergent could be removed. After this cyclodextrin addition protein aggregates could be observed. The grid also showed much less background. The detergent could be removed that way, but seemingly too fast for the protein to form crystals or even to integrate in a lipid bilayer.

The results from the experiments performed with DDM yielded better results, but also in these samples the strong background indicated that not all the detergent could be removed. The background was much weaker after spinning the samples for 20’ at 100’000 g and re-suspending the pellet. In these samples, aggregates and some non crystalline vesicles could be observed (see figure 2.6 for an example of a big vesicle).

Some experience about the behaviour of MBCD could be gained: The spinning step at the end of the experiment to get rid of the surplus of MBCD in the sample was used in further experiments and the calculation of how much cyclodextrin one needs to remove a certain amount of detergent was improved.

No further experiments on AQP1 were performed after this, since more abundant sources of other proteins (OmpF and SoPiP2; 1) were available to further elaborate the cyclodextrin method.

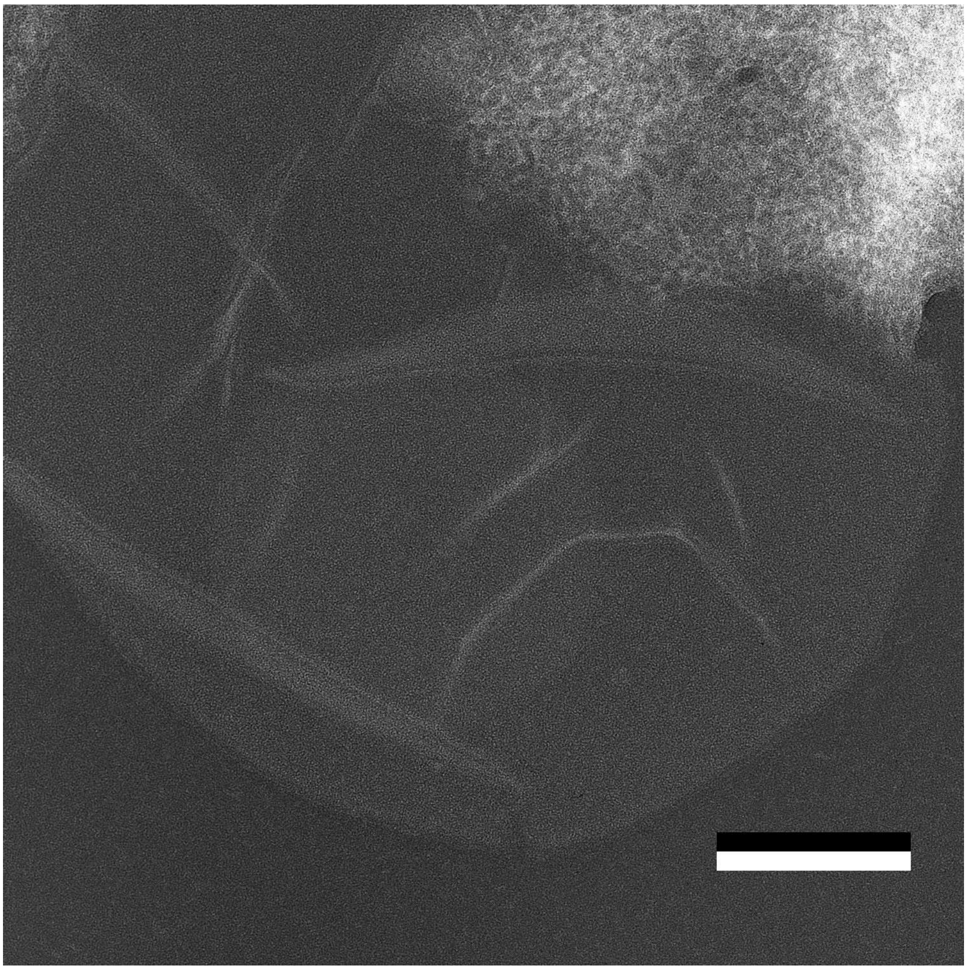


Figure 2.6: AQP1 Vesicle.
Example of a big AQP1 vesicle. The LPR used for this experiment was 1 and the reconstitution time was 12 h. The scale-bar represents 200 nm.

2.2.5.2 AQP8

A collaboration of Wanda Kukulski with Maria Karlsson from the Lund University in the context of protein reconstitution gave us access to the membrane protein AQP8 [29]. The cyclodextrin experiments were only preformed with the leftovers of other experiments and thus only very little protein was available for the cyclodextrin experiments. Only few experiments could be performed.

Materials and Methods The protein was expressed and purified by Maria Karlsson. The protein concentration was 0.6 mg/ml. *E.Coli* lipid, a mix of cardiolipin and DOPG (2:1) and DMPC were used to get LPRs of 0.3 and 0.8. All three lipids were solubilized in 1 % DM. The ternary mixtures were incubated for 1 h at room temperature.

The detergent was removed as described in [46]. 400 μl of a 1% MBCD solution was added over 96 h to 40 μl of ternary mixture. The specimens for negative stain electron microscopy were prepared as explained in section 2.2.5.1. Micrographs were taken on a Hitachi H-8000 transmission electron microscope operated at 200 kV using magnifications of 5000 and 50’000.

Results and Discussion Most of the performed experiments yielded aggregates. At the higher LPR of 0.8, some vesicles and aggregates of vesicles could also be observed. At the LPR of 0.3 strong background was observed on the grids.

Too few experiments were made to come to any reasonable conclusion. The choice to use three different lipids, was mainly made to gain some experience in the variation of lipids. But since only two conditions (LPR 0.3 and LPR 0.8) were tested, the results are not really significant.

AQP8 reconstitution trials were continued using dialysis and recently first crystals have been observed.

2.2.5.3 GalP

The presented results on GalP can be seen as part of a bigger screening process as no crystallization conditions of GalP were known.

Materials and Methods Membranes from *E. coli* strain JM1100 (pPER3) over-expressing GalP were kindly provided by P.J.F. Henderson and the purification of the protein was done by Thomas Kaufmann as described in [25].

Each of the GalP elutions 1 (0.3 mg/ml), 2 (0.4 mg/ml) and 3 (0.4 mg/ml) was dialyzed 72h at 4°C against a separate buffer to have exactly defined starting conditions: Elution 1 was dialyzed against 20 mM MES, pH 6, 0.01% DDM, 5% glycerol, 50 mM NaCl and 2mM NaN_3; elution 2 against 20 mM Tris, pH 7.5, 0.01% DDM, 5% glycerol, 50 mM NaCl and 2mM NaN_3; and elution 3 against 20 mM Tris, pH 9, 0.01% DDM, 5% glycerol, 50 mM NaCl and 2mM NaN_3.

DMPC (in DDM) was added to cover LPRs of 1, 0.6, 0.4 and 0.2 (resulting in a DDM concentration of maximal 0.15%). The ternary mixtures were incubated for 2h at room temperature.

The detergent was removed as described in the presented paper [46]. Experiments over different time ranges were performed adding MBCD solution of a concentration high enough to remove all the detergent from the mixture. LPR 1 experiments were performed over 2h and 6h and LPR 0.6, 0.4 and 0.2 experiments over 4h. In all experiments 20 μl of a 1.5% MBCD solution were added to 20 μl ternary mixture.

The specimens for negative stain electron microscopy were prepared and stained as explained in section 2.2.5.1. Micrographs were taken on a Hitachi H-7000 transmission electron microscope at operated 100 kV and on a Hitachi H-8000 transmission electron microscope operated at 200 kV using magnifications of 5000 and 50’000.

An optical bench with a laser beam was used to evaluate the diffraction quality of the micrographs.

Results and Discussion All the reconstitution experiments resulted in the formation of vesicles. The 2h experiments with an LPR of 1 yielded the smallest vesicles, which had also the tendency to aggregate. The 6h experiment with an LPR of 1 resulted in bigger vesicles (see figure 2.7) and, in the pH 6 sample additionally some bigger sheet-like structures. No diffraction spots could be observed in the optical diffraction of these images.

The LPR 0.6 and 0.4 experiments resulted in an increased size of the vesicles, with the best results (concerning the size of the vesicles) at pH 6. No diffraction spots were observed in the optical diffraction of these images.

At LPR 0.2 there was a coexistence of aggregates and vesicles. This coexistence was most likely an effect of the imperfect mixing of the sample and thus of the inhomogeneity of the whole sample. This issue is addressed in the new device under construction described in section 2.2.4.

The results from this experiments showed some preference for the pH 6 conditions. Longer experiments, and thus a slower addition of MBCD for detergent removal, would be worth considering in a continuation of the project.

2.2.5.4 NaChBac

To gain insight on the function of the sodium channel NaChBac, Thomas Braun initiated experiments to create 2D crystals suitable for structure determination. The project was taken over by Simon Bernèche and Wanda Kukulski. Most of the experiments were performed using the dialysis method and where performed mainly by Wanda Kukulski.

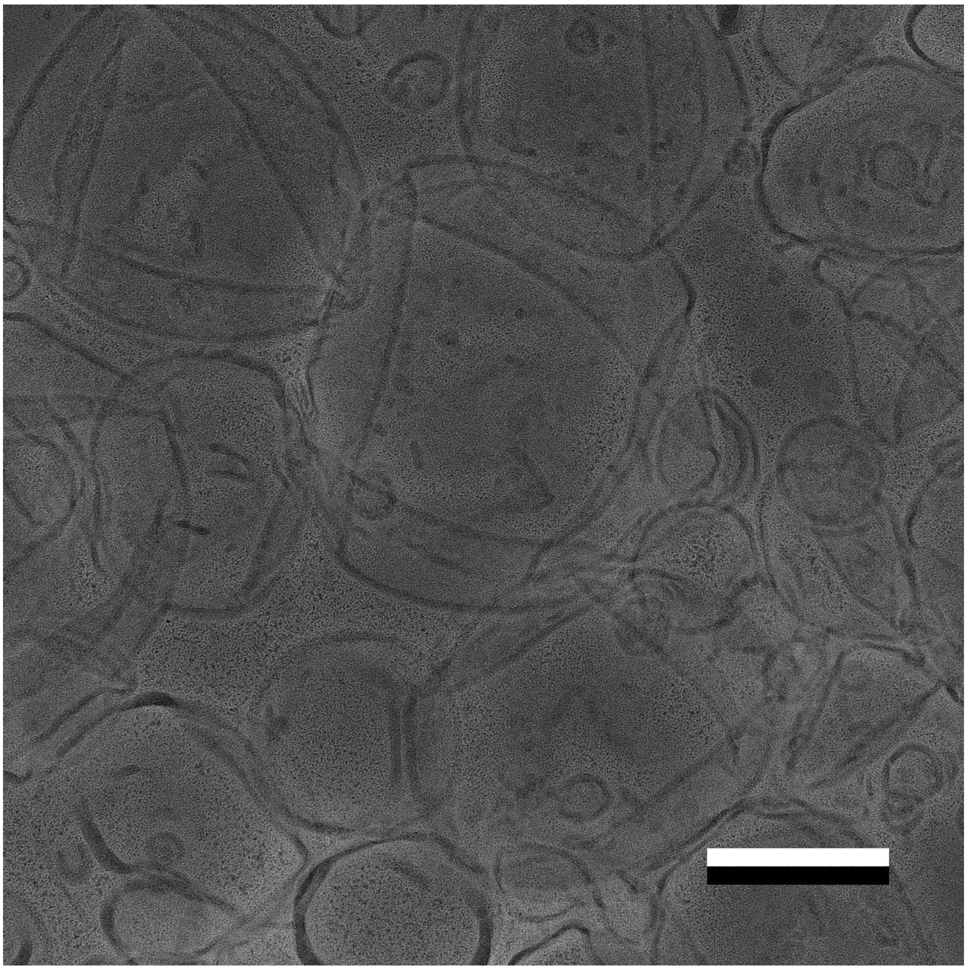


Figure 2.7: GalP Vesicles

GalP vesicles from a 6 h experiment using an LPR of 1 and a pH of 9. The scale-bar represents 200 nm.

Materials and Methods NaChBac was expressed by Xiaodan Li and purified by Thomas Braun and Simon Bernéche. The protein concentration was determined using the BCA-assay from Pierce [47], indicating a protein concentration of 0.5 mg/ml.

A lipid mixture of 5 mg/ml DOPS:DOPC (70:30) in 1% DDM was added to the protein solubilized in 0.4% DDM to get mixtures of LPR 0.4, 0.5, 0.6, 0.7 and 0.8. The ternary mixtures were incubated for 2h at room temperature.

The detergent was removed as described in [46]. 600 μl of a 0.22% MBCD solution were added over 144h to 40 μl of ternary mixture.

The specimens for negative stain electron microscopy were prepared as explained in section 2.2.5.1. Micrographs were taken on a Hitachi H-7000 transmission electron microscope operated at 100 kV using magnifications of 5000 and 50’000.

Results and Discussion In all tested conditions only vesicles could be observed (see figure 2.8). At lower LPRs the vesicles were bigger, but none of the samples showed crystals. No aggregates were observed in any of the conditions, leading to the assumption that the tested LPRs were too high.

Many more conditions were screened using the dialysis method, including a big range of lipids, detergents, salts and LPRs. None of the tested conditions yielded crystals. The high number of tested conditions indicates that the protein has either very special crystallization conditions in a narrow range or that the protein is not suited for 2D crystallization.

2.2.5.5 Rho and YCD

In a collaboration with Xiaodan Li we tried to crystallize samples of the two proteins, Rho and YCD, with the dialysis (Wanda Kukulski and I) and the cyclodextrin method. Wanda also performed single particle experiments to asses the quality of the protein samples.

Materials and Methods We got the proteins from Xiaodan Li in concentrations of 0.7 mg/ml for Rho and 2.7 mg/ml for YCD. To have exactly defined starting conditions the protein was dialyzed overnight at 4°C against a buffer containing detergent. The buffer used for Rho contained 20 mM Caps, pH10, 100 mM NaCl, 2mM NaN_3 and 0.03 % DDM. The buffer YCD was dialyzed against contained 20 mM Tris, pH 8, 100 mM NaCl, 2mM NaN_3 and 4 mM LDAO.

E. Coli lipid was added to Rho to make ternary mixtures of LPRs 0.3 and 1. For YCD *E. Coli* lipid, a mix of DMPC:DOPS (70:30) and soybean lipids was added to the protein to get LPRs of 0.3 and 1. The ternary mixtures were incubated for 2h at room temperature.

The detergent was removed using MBCD as described in [46]. For Rho 35 μl of a 1.5% MBCD solution was added over 24h to 35 μl of ternary mixture. For YCD 300 μl of a 2.6% MBCD solution was added over 72h to 40 μl of ternary mixture.

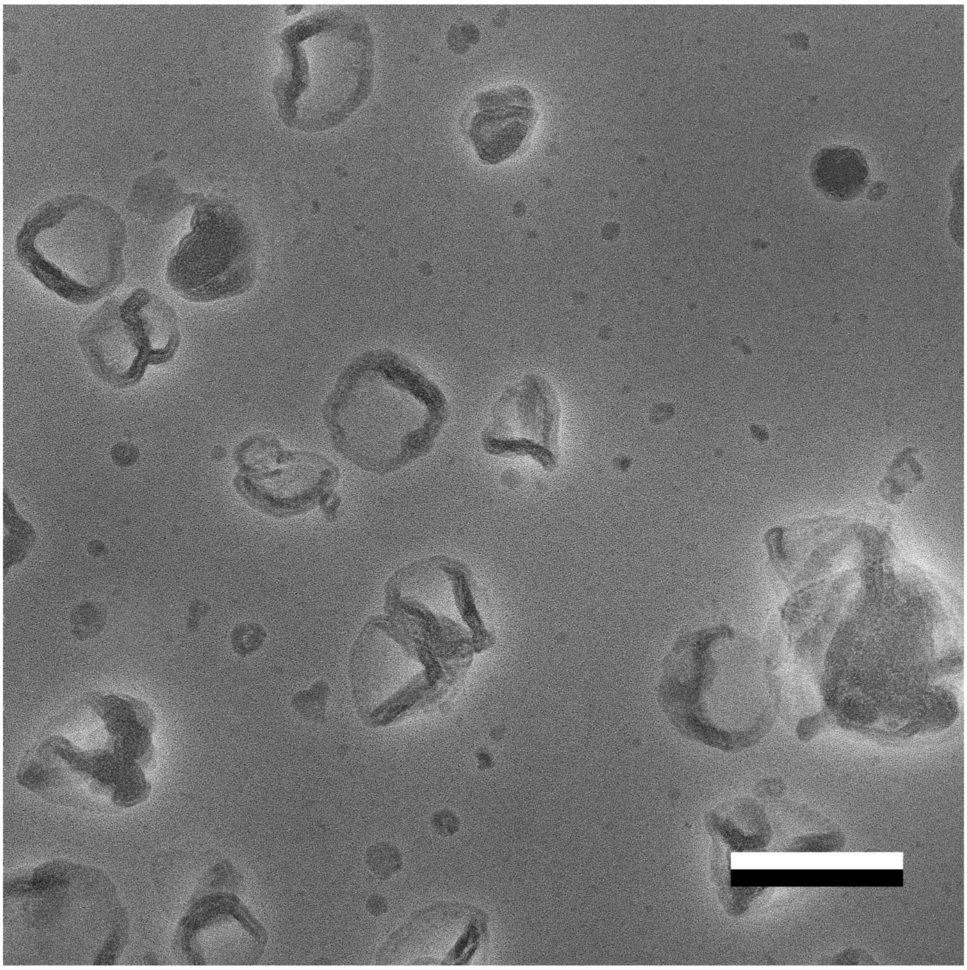


Figure 2.8: NaChBac Vesicles
NaChBac vesicles resulting from an experiment at LPR 0.7. The scale-bar represents 200 nm.

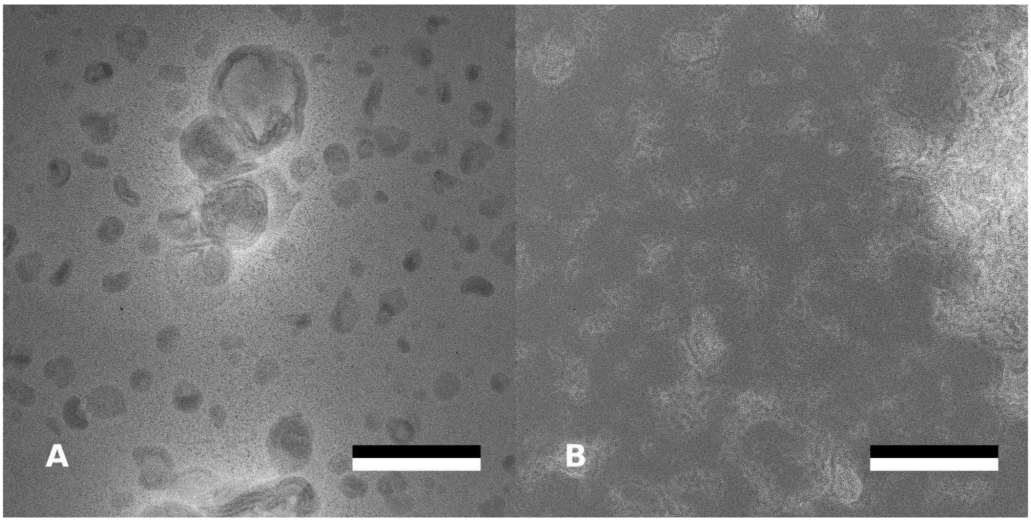


Figure 2.9: Rho and YCD Vesicles
A) Rho vesicles resulting from a 24 h experiment using an LPR of 0.3. B) YCD vesicles and aggregates resulting from a 72 h experiment using an LPR of 0.3 (with *E. Coli* lipids). The scale-bars represent 200 nm.

The specimens for negative stain electron microscopy were prepared as explained in section 2.2.5.1. Micrographs were taken on a Hitachi H-8000 transmission electron microscope operated at 200 kV using magnifications of 5000 and 50’000.

Results and Discussion The reconstitution experiments for Rho resulted in multilayered and normal vesicles for LPR 1 as well as for LPR 0.3 (see figure 2.9 A). At the LPR of 0.3 additionally aggregates were observed. The amount of protein we got was very limited, not allowing to perform more experiments. The single particle analysis performed by Wanda showed that the protein sample was not really pure.

The reconstitution experiments for YCD showed similar results (see figure 2.9 B). No preference for any of the lipids could be observed. Most samples contained small vesicles and at the lower LPR of 0.3, some aggregates were observed. In some cases big parts of the plastic film of the grids was destroyed, leading to the assumption that the detergent was not removed completely.

Crystallization experiments using dialysis did not show much better results and the limited amount of protein and the impurities prevented further crystallization experiments.

2.2.5.6 Conclusions

The experiments were performed in parallel with the systematic search for crystallization conditions for KdgM, KdgN and NanC and the development of the novel crystallization method using cyclodextrins, whenever protein leftovers were available from other projects. This way not too much protein was wasted in experimenting with shorter time ranges and different MBCD concentrations. All these projects show that without a systematic screening of conditions for crystallization, not much success in the crystallization can be expected. The benefits of these experiments are the gained experience in handling the setup and some insight about the kinetic processes in detergent removal using MBCD.

2.3 Outer Membrane Porins KdgM, NanC and KdgN

2.3.1 Prior Work

At the time I started with the crystallization experiments for KdgM, Hervé Remigy had tested already several conditions. I started the project with sample preparation (negative stain) and electron microscopy of the last conditions Hervé had prepared. In these samples some very small crystals were found. The crystals were smaller than 200 nm and showed some weak diffraction spots, similar to crystals Hervé Remigy found earlier in other conditions (personal communication).

Once better KdgM crystals were obtained, I started crystallization screens for NanC and KdgN based on the conditions found for KdgM. No crystallization experiments were performed with these two proteins from G. Condemine prior to my work.

2.3.2 Initial Screening Using MBCD

In the first time, most of the crystallization experiments with KdgM were performed using the cyclodextrin method for detergent removal [46]. I had gained enough experience with this technique with the membrane proteins OmpF, SoPiP2;1 and the proteins presented in section 2.2.5 to have reproducible results, comparable with results from the dialysis method. The disadvantage of the experimental setup in having inhomogeneous samples due to the imperfection of the mixing device (see section 2.2.4), turned out to be an advantage in the sense that *different* conditions could be tested at the same time. The differences in concentration arising from the inhomogeneities were small, but nevertheless vesicles, aggregates and crystalline areas could be found in the same condition. I invested a lot of time in the screening with the electron microscope of these samples and could thus find even small crystalline areas at borders of aggregates or crystalline tubes in samples containing mainly vesicles of approximately the same size.

The crystallization experiments using MBCD to remove detergent from the ternary mixtures are mentioned in the 3 porins paper presented in section 2.3.6 only very briefly, therefore a more extensive presentation of the experiments will be presented here.

2.3.2.1 Materials and Methods

To search for crystallization conditions several different detergents, LPRs, lipids and salts were tested. The time ranges of the experiments (i.e. the kinetics of detergent removal) and the amount of MBCD was also varied to gain more insight into the reconstitution kinetics.

The dialysis against a buffer to have controlled starting conditions, the preparation of the ternary mixture and the experimental setup to remove the detergent is described in [46]. The protein concentrations were measured using a BCA assay from Pierce [47] and the detergent concentrations were measured using the detergent-box described in [25].

The conditions tested for the different proteins were as follows:

KdgM Ternary mixtures containing different salts, different LPRs and different detergents were prepared.

The salts used were 100 mM, 200 mM, 300 mM and 400 mM NaCl, to some of the 400 mM NaCl conditions 10 mM $MgCl_2$was added.

DMPC was used at 5 mg/ml and at 10 mg/ml to prepare LPRs of 0.05, 0.1, 0.12, 0.14, 0.15, 0.16, 0.18, 0.2, 0.4, 0.6, 0.8 and 1.0.

The detergents used were LDAO (0.25%, 0.5% and 1%), TX-100 (0.4% or 1%) and $C_{12}E_8$ (0.4% and 1%).

The initial buffers contained 2 mM $NaN_3$and 20 mM tris at pH 9. The amount and concentration of the cyclodextrin solutions was chosen according to the experiment duration. The experiment durations were 2h, 4h, 6h, 16h, 72h and 144h. 5% of glycerol was added to some of the TX-100 and $C_{12}E_8$ experiments.

KdgN Ternary mixtures containing 200 mM NaCl and LPRs of 0.05, 0.1 and 0.15 with DMPC were used. The detergent used was 0.25% LDAO. The initial buffers contained 2 mM $NaN_3$and 20 mM tris at pH 9. The amount and concentration of the cyclodextrin solutions was chosen according to the experiment duration. The experiment durations were 72h and 96h.

NanC Ternary mixtures containing 200 mM NaCl and LPRs of 0.1 and 0.15 with DMPC were used. The detergent used was 0.25% LDAO. The initial buffers contained 2 mM $NaN_3$and 20 mM tris at pH 9. The experiment durations was 72h and 300μl of a 1% MBCD solution was added to 40μl over 72 h.

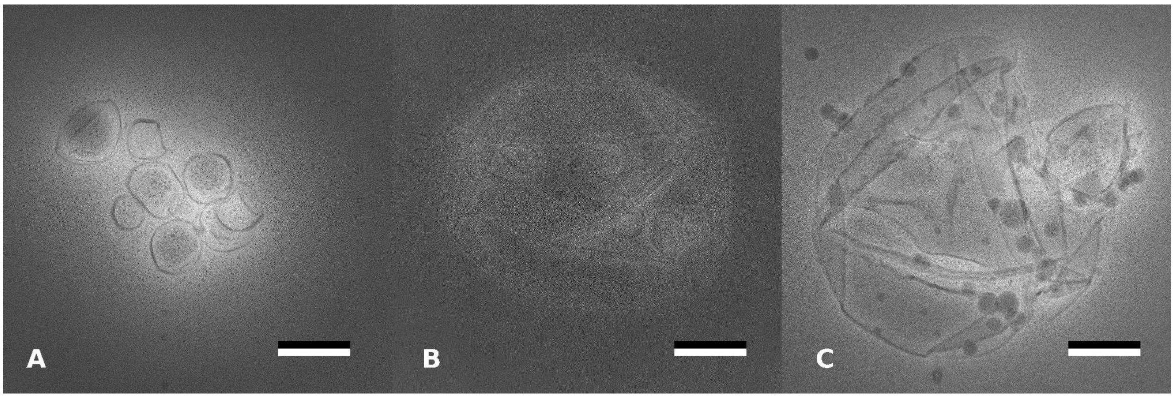


Figure 2.10: Vesicles
A) Small vesicles of a size around 100 nm. This image was taken from a 96 h KdgN experiment. B) and C) Bigger vesicles can have crystalline areas, can be densely packed with protein or can be without protein insertion. B) was taken from a 4h KdgM experiment at an LPR of 0.2 and C) from a 96 h KdgN experiment at an LPR of 0.1. The scale-bars correspond to 200 nm.

2.3.2.2 Results and Discussion

The screening of crystallization conditions with the electron microscope showed different kinds of objects on the grids: small vesicles, bigger vesicles, tubes, sheets, aggregates of protein or aggregates of vesicles, background and stain artifacts. These objects were usually not populating the whole grid in a homogeneous way and thus often a combination of various different objects could be observed.

In the following there will be a description and examples of the different objects that were found in the experiments listed above.

Small Vesicles Small vesicles (of a size of 100 nm and less) often appeared when the protein did not reconstitute into the lipid bilayers or when there was a much to high lipid concentration in the samples. This could be seen in combination with protein aggregates when the LPR was too low or in other cases without protein aggregates when the LPR was too high (LPR 0.4 -1).

Other factors influencing the formation of these small vesicles were the choice of detergent (e.g. using LDAO yielded small vesicles, but TX-100 resulted only in the formation of aggregates at the same conditions), and the experiment duration (longer experiments usually resulted in the formation of bigger vesicles, indicating the importance of the kinetics for the reconstitution).

Figure 2.10A shows an example of small vesicles. The image shows the result of a KdgN experiment, the LPR was 0.1 using DMPC and a buffer containing 20 mM tris, pH 9, 200 mM NaCl, 1 mM $MgCl_2$, 2mM NaN_3 and 0.25 % LDAO. The experiment duration was 96h.

Bigger Vesicles Vesicles of a size bigger than 500 nm were usually a good indication of having success in the reconstitution of the protein into the membrane. Aggregates were rarely observed in combination with these bigger vesicles and the optical diffraction often showed a dense packing of protein. In some cases even crystalline areas of low order could be observed.

In the experiments with KdgM, KdgN and NanC, bigger vesicles often appeared at low salt concentrations (lower then 200 mM NaCl) and using LDAO as detergent.

In the figure 2.10 two examples of such bigger vesicles are shown. Figure 2.10B shows the result of a KdgM experiment over 4 h with an LPR of 0.2 using a buffer containing 100 mM NaCl and figure 2.10C a KdgN experiment over 96 h with an LPR of 0.1 using a buffer containing 200 mM NaCl. LDAO and DMPC were used in both experiments. Vesicles of a size up to 1 μm were observed.

Tubes The tubes found in my cyclodextrin experiments were usually crystalline and had about the same size as the bigger vesicles. Their shape varied from long thin tubes (Fig.2.11B) to short wide tubes (Fig. 2.11A) looking similar to the big vesicles. In the cyclodextrin crystallization trials with KdgM, only few conditions yielded crystals. From these conditions only in the longer reconstitution experiments crystalline tubes could be observed. These tubes were smaller than the tubes found later in similar conditions using the dialysis method (see the Paper 2.3.6 for more details). In figure 2.11 you can see crystalline tubes of a size of 300 to 400 nm. These KdgM crystals were observed at an LPR of 0.05 with DMPC using a buffer containing 20 mM tris, pH 9, 200 mM NaCl, 2mM NaN_3 and 0.5 % LDAO. The initial protein concentration was 1.8 mg/ml. The experiment duration was 144 h and 600 μl of a 1.3% MBCD solutions was added to 40 μl ternary mixture.

Sheets Sheets are flat bilayer areas. In most of the observed cases in the cyclodextrin experiments they could be seen at the borders of aggregates and were rather small (see also the crystal in figure 2.16A). This resulted from the inadequate mixing, leading to concentration gradients from the faster detergent removal at the place were the cyclodextrin was added to the probe.

Bigger sheets were observed only in few cases. These sheets were densely packed with protein, but did not show any diffraction spots in the optical diffraction of the images. In figure 2.12 one example of a KdgM sheet of a size of more than 2 μm can be seen. The experiment duration of this experiment was 16 h, the LPR 0.15, and the buffer contained 400 mM NaCl. The other areas on the same grid contained mainly aggregates.

Aggregates of Protein Aggregates of protein formed at very low LPRs (usually lower than 0.1) as there was not enough lipid to form proteoliposomes. When the detergent was removed from these samples, the protein molecules started to aggregate. Aggregates could also be observed in other conditions when the protein could not insert into the lipid bilayer. This was often the case, when the

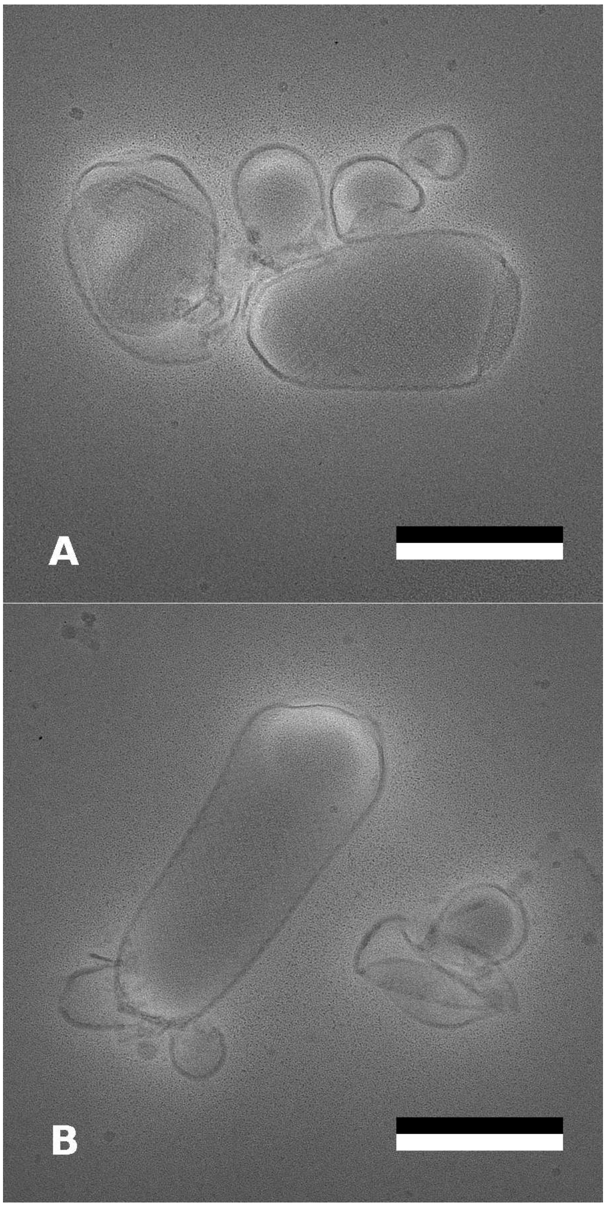


Figure 2.11: Tubes
The tubes found in our experiments were usually crystalline. The images were taken from a KdgN experiment at an LPR of 0.05. Tubes can have different proportions: In A) the tubes are almost vesicle-shaped, whereas the tubes in B) are much longer than wide. The scale-bars indicate 200 nm.

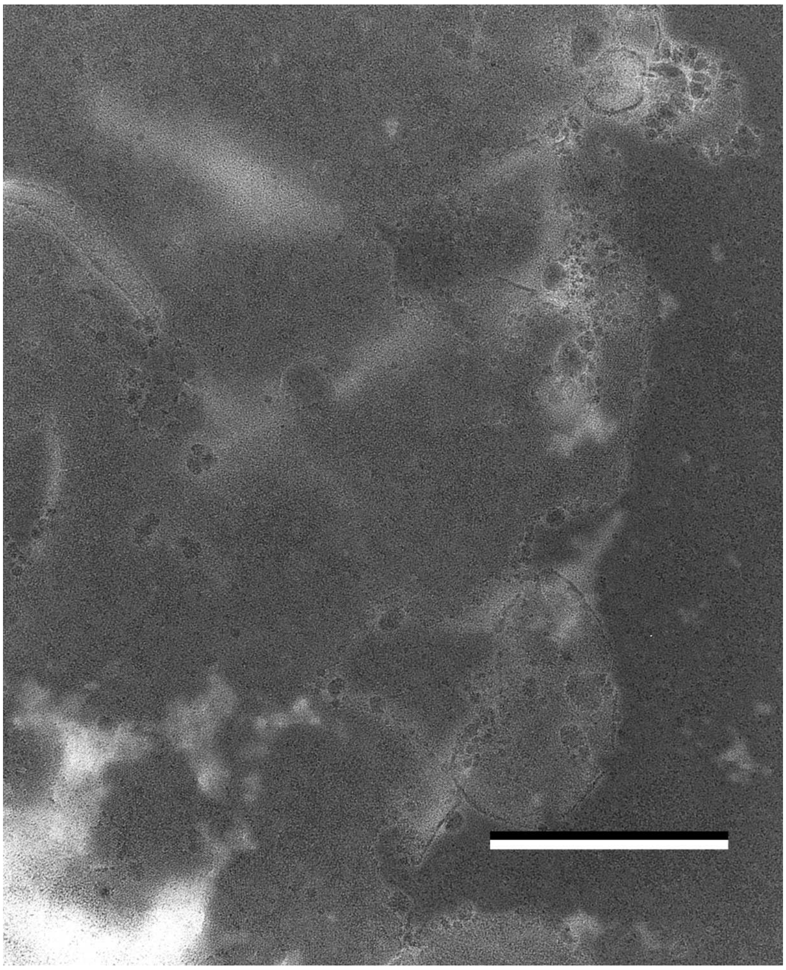


Figure 2.12: KdgM Sheet.
This image of a KdgM sheet was taken from a sample of a 16h experiment with an LPR of 0.15. The rest of the grid contained mainly aggregates. The scale-bar indicates 400 nm.

detergent removal was too fast (experiments lasting less than 12h). Certain detergents like $C_{12}E_8$and TX-100 usually yielded protein aggregates, also at higher LPRs.

Aggregates of various sizes could be observed, from very small objects to big structures spanning several grid-holes at once. The size of the aggregates was mainly dependent on the overall protein concentration, higher protein concentrations yielding bigger aggregates than more diluted samples resulting for example from the dilution with a high volume of cyclodextrin solution.

Aggregates were usually observed together with other objects, such as (small) sheets or even vesicles and tubes. This can be explained in the imperfection of the experimental setup concerning the mixing.

Figure 2.13 shows examples of aggregates: A) is from a 16 h KdgM reconstitution, LPR 0.2, buffer containing 400 mM NaCl and B) from a KdgM reconstitution, LPR 0.15, buffer containing 400 mM NaCl and 10 mM $MgCl_2$.

Aggregates of Vesicles Aggregates of vesicles were found very often under conditions in which the lipid concentration was high, most of the time when only a small volume of cyclodextrin solution was added to the sample. The choice of detergent also had an influence on the aggregation of the vesicles: With $C_{12}E_8$, for example, most of the experiments resulted in the formation of aggregates.

Aggregates of vesicles are often difficult to distinguish from protein aggregates, since both types of aggregates stain very dark in negative stain.

In figure 2.13 C an example of such an aggregate of vesicles is shown. The KdgM ternary mixture with an LPR of 0.1, was prepared using $C_{12}E_8$ as detergent. The experiment duration was 6h and only 30 μl of a 5% MBCD were added to 30 μl of ternary mixture.

Background Some of the samples showed a strong background and in some of the cases nothing else was visible on these sample grids. The strong background had various origins: In the first experiments little was known about the concrete behaviour of cyclodextrin in removing detergents from the ternary protein-lipid-detergent mixtures as the detergent titration measurements performed by Thomas Kaufmann were not yet done at that stage of the experiments. This lack of knowledge is the reason why, in some of the samples, not all the detergent was removed. The remaining detergent lead to the destruction of the parlodion film of the grids in some places and in the presence of lots of solubilized lipid or protein in other places, giving a strong background. Sometimes also the dilution machine failed to work properly, blocking the addition of MBCD and therefore the complete removal of detergent. Another source of background was the uranyl acetate stain, when it was not blotted properly.

In figure 2.14 you can see an example of an image with strong background. The image was taken from a KdgM sample where the machine adding MBCD was blocked and thus not all detergent could be removed.

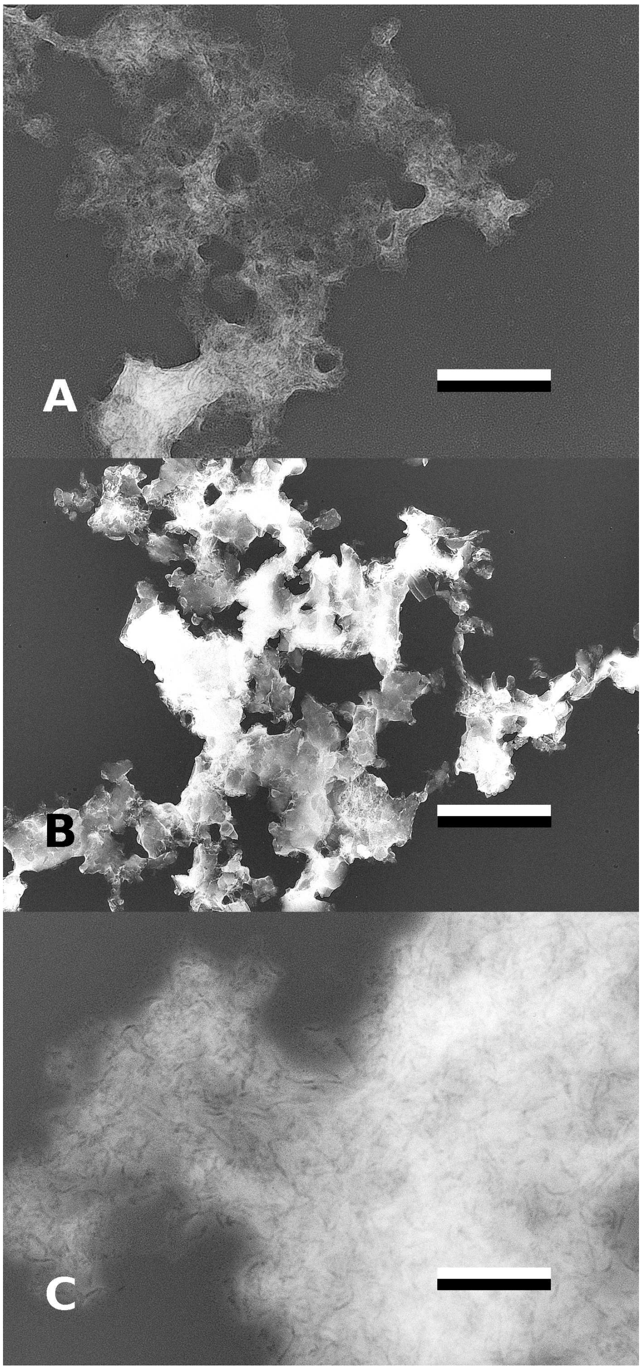


Figure 2.13: Aggregates

Examples of big KdgM aggregates, A) is a picture taken from a 16 h experiment at an LPR of 0.2, B) from a 72 h experiment with an LPR of 0.15. C) shows an aggregate of vesicles resulting from a 6 h experiment with an LPR of 0.1, using $C_{12}E_8$. The amount of MBCD solution added to the ternary mixture was very low. The scale-bars indicate 200 nm.

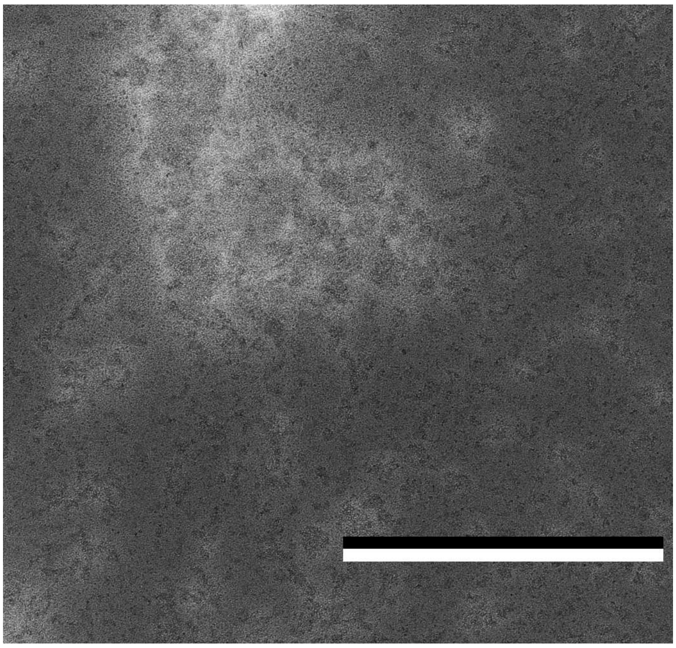


Figure 2.14: Background
Strong background usually arises from an incomplete detergent removal and thus protein and lipids remaining solubilized. This image was taken from a KdgM sample, where the machine to add MBCD to the sample stopped during the experiment and thus not all detergent could be removed. The scale-bar indicates 400 nm.

Not a lot of images were taken when the grids contained mainly background. The existing pictures were taken at the beginning of my screening experience, at a time were I was not sure about the quality of the different objects that can be seen on the EM-grids.

Stain Artifacts The uranyl acetate used to stain the samples is blotted using a filter paper and thus only the stain attached to the objects should remain on the grid and give contrast where there are organic compounds. The blotting was not always perfect and so on some regions of the grid the remaining uranyl acetate dried and resulted in various shapes visible on the grid. Another source of stain artifacts was the not always optimal quality of the stain, having crystalline uranyl acetate bits in the solution, visible also on the grids.

Stain artifacts are nothing one would normally take pictures of, but since they sometimes look like objects one is searching for, or because they have ’nice’ shapes, some pictures were taken of these artifacts. As an example you can see a rather big object on figure 2.15.

Crystals The first KdgM crystals that were found in the cyclodextrin experiments were smaller than 200 nm and could be observed at the border of aggregates, forming sheet- or vesicle-like structures. In figure 2.16 A, crystalline areas of KdgM can be seen at the border of an aggregate.

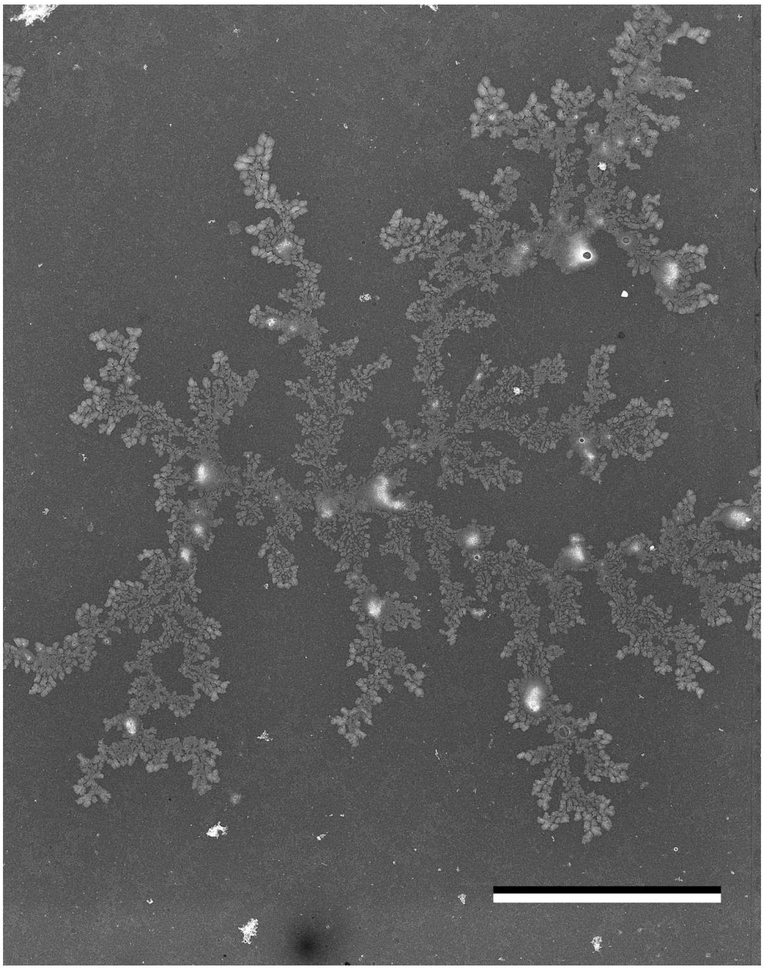


Figure 2.15: Stain Artifacts
This image was taken from a grid of a KdgM experiment, but no protein or lipid is visible on this image. The scale-bar indicates $4\mu m$.

Crystals could be found under different conditions using the cyclodextrin method. Since more conditions were tested for KdgM than for NanC and KdgN, different conditions could be found where KdgM crystals formed. Only one crystallization condition could be found for NanC and no condition was found where KdgN formed crystals.

KdgM The first condition in which crystals of KdgM were found was prepared using 400 mM NaCl, 0.25 % LDAO and an LPR of 0.15. 200 μl of a 0.5% MBCD solution were added to 20 μl ternary mixture over 16 h. Figure 2.16 A shows crystalline areas of this sample.

Other conditions yielding crystals were prepared using 200 mM NaCl, 1 % LDAO and an LPR of 0.15. 600 μl of a 1.3 % MBCD solution were added to 40 μl ternary mixture over 144h. In this sample crystalline vesicles could be found (see figure 2.16 B), but their diffraction quality was rather low.

Crystals of a better quality forming small (300 - 400 nm) tubes could be found using 200 mM NaCl, 0.5% LDAO and an LPR of 0.05. 600 μl of a 1.3 % MBCD solution was added to 40 μl ternary mixture over 144h. Images from this samples can be seen in the figure 2.11.

NanC The condition were NanC crystals formed contained 200 mM NaCl and 0.25 % LDAO. DMPC was used to reach an LPR of 0.15. The detergent was removed by adding 300 μl of a 1 % MBCD solution over 72 h to 40 μl ternary mixture. The NanC crystals produced with cyclodextrin could be found in sheet-like structure as can be seen in figure 2.16 C. Not the whole sheet was crystalline and the diffraction spots were weak.

Conclusions At the beginning of the cyclodextrin experiments with the three porins KdgM, KdgN and NanC only initial experience concerning the kinetics of the detergent removal was gained from experiments with other proteins. This explains the big number of experiments over short time ranges of less than 12 h. As more experience about the kinetic behaviour was gained, more of the longer experiments were performed. At the beginning of the MBCD experiments, only KdgM was available. Latter on, KdgN and NanC could also be used for experiments. At that time some crystallization conditions of KdgM were already known and were refined using the dialysis method.

In order to screen as many conditions as possible, KdgN and NanC were also screened using the same dialysis buffers. The setup of the dilution machine allowing only 6 to 8 experiments to be performed in parallel made it more efficient to screen a high number of conditions using the dialysis method. This is the reason why less experiments using the cyclodextrin method were performed.

KdgM crystallized before the first experiments with KdgN and NanC were performed. The first conditions to test for these two proteins were therefore the crystal yielding conditions of KdgM. NanC crystallized in conditions similar to KdgM, whereas no crystallization conditions could be found for

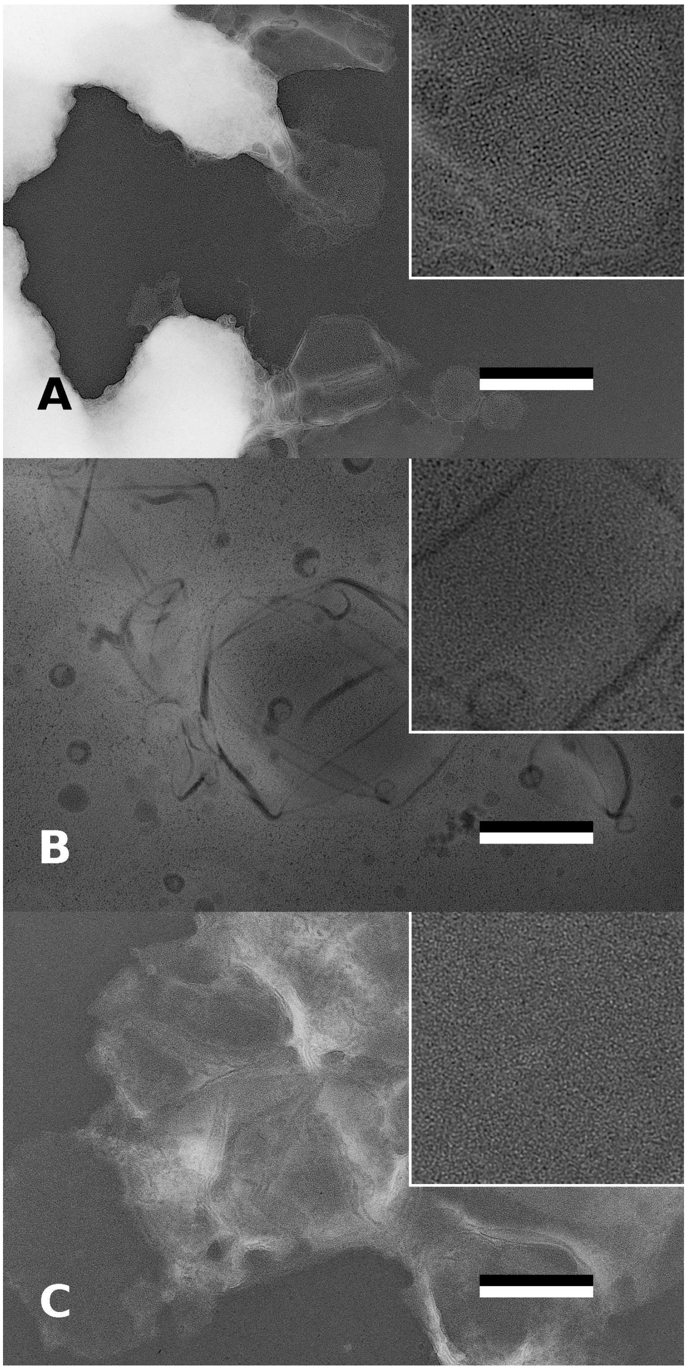


Figure 2.16: KdgM and NanC Crystals

A) Crystal of KdgM at an LPR of 0.15, after a reconstitution lasting 16 h. The initial buffer from this sample contained 400 mM NaCl and 0.25 % LDAO. B) Crystalline vesicle of KdgM at an LPR of 0.15, after a reconstitution time of 144 h. The initial buffer contained 200 mM NaCl and 1 % LDAO. C) Crystalline sheet of NanC at an LPR of 0.15, after a reconstitution lasting 72 h. Not the whole area is crystalline and the diffraction spots are weak. The scale-bars indicate 200 nm. The insets are magnified 3 times.

KdgN using the cyclodextrin method. Later on, in the dialysis experiments, quite different crystallization conditions were found for KdgN (see section 2.3.6 for more details).

2.3.3 Systematic Screening Using Dialysis Buttons

After having found crystallization conditions for KdgM, systematic crystallization screenings with the more reproducible dialysis method appeared to be sensible. This screening could be done in parallel with all the three proteins, giving the possibility to search for more crystallization conditions for NanC and KdgN. In the search of conditions using dialysis also very different conditions were tried out, according to the experience from other protein crystallization experiments or according to conditions used in the purification of the proteins [5].

An extensive summary of the conditions tested using dialysis buttons can be seen in the paper, presented in section 2.3.6.

2.3.4 Quality Control with the Electron Microscope

To see whether the crystallization experiments were successful or not, the resulting samples needed to be checked using the electron microscope. For this screening, negative staining of the samples was used. Negative stain has the advantages of having a high contrast and of being stable in the electron beam for an extensive search. The disadvantage of having lower resolution is not relevant in the screening process, since the data collection is performed using cryo electron microscopy.

The screening of cyclodextrin-prepared samples is very time consuming, since it was often necessary to look at a bigger part of the grid because the samples were not always homogeneous. This extensive search of the single grids allowed me to find even very small crystalline areas at borders of aggregates or between non-crystalline vesicles.

2.3.5 Contributions

My contribution to the presented work included the preparation and screening of the samples prepared with cyclodextrin with the help and under the initial supervision of Hervé Remigy and the dialysis experiments (with guidance of Hervé Remigy) including the screening of negative stained samples using the electron microscope. The cryo electron microscopy was done mostly by Mohamed Chami (KdgM and NanC) and Andreas Schenk (KdgN) together with me. I developed and scanned the images for later image processing. I performed the image processing using MRC and the new IPLT-GUI and was supported by Andreas Schenk. Ansgar Philippsen was very supportive in the process of correcting and assembling the paper.

2.3.6 The Porin Paper

A paper on the crystallization and the image processing of the 3 porins was submitted to JSB in Mai 2007 under the title "Projection Maps of Three Members of the KdgM Outer Membrane Protein Family". The following manuscript is the submitted version.

Projection Maps of Three Members of the KdgM Outer Membrane Protein Family

Gian A. Signorell[1], Herve W. Remigy[1], Mohamed Chami[1], Guy Condemine[2], Andreas D. Schenk[1], Ansgar Philippsen[1], and Andreas Engel[1*]

[1] Maurice E. Müller Institute for Structural Biology, Klingelbergstr. 70, 4056 Basel, Switzerland

[2] Université de Lyon, F-69003, France; Université Lyon 1, F-69622; INSA-Lyon, Villeurbanne, F-69621; CNRS, UMR5240, Unité Microbiologie Adaptation et Pathogénie, F-69622

*Corresponding author
Prof. Andreas Engel
M.E. Müller Institute for
Structural Biology
Biozentrum, University of Basel
Klingelbergstrasse 50/70
CH-4056 Basel / Switzerland

Phone: +41 61 267 22 6
Fax: +41 61 267 21 09

Abstract

We present the projection structures of the three outer membrane porins KdgM and KdgN from *Erwinia chrysanthemi* and NanC from *Escherichia coli*, based on 2D electron crystallography. A wide screening of 2D crystallization conditions yielded tubular crystals of a suitable size and quality to perform high-resolution electron microscopy. Data processing of untilted samples allowed us to separate the information of the two crystalline layers and resulted in projection maps to a resolution of up to 5 Å. All three proteins exhibit a similar putative β-barrel structure and the three crystal forms have the same symmetry. However, there are differences in the packing arrangements of the monomers as well as the densities of the projections. To interpret these projections, secondary structure prediction was performed using β-barrel specific prediction algorithms. The predicted transmembrane β-barrels have a high similarity in the arrangement of the putative β-strands and the loops, but do not match those of OmpG, a related protein porin whose structure was solved.

Keywords

Outer membrane porins; Cryo electron microscopy; 2D crystallization; Structure prediction;

Introduction

KdgM, NanC and KdgN are members of the outer membrane pore forming protein family, usually referred to as porins. Several reviews describe the structure and function of these proteins and their common structural feature, the β-barrel (Schulz, 1996; Schirmer, 1998; Koebnik et al., 2000). The presented proteins are about 220 amino acids long and are thus the smallest porins described up to now.

KdgM is the major outer membrane protein of the gram-negative bacterium *E. chrysanthemi*. This bacterium secretes pectinases to degrade the pectic polymers of plant cell walls, and hence to use these degradation products as a carbon source for growth. Genetic and electrophysiological properties of KdgM have been characterized by Blot et al. (2002). They demonstrate that KdgM synthesis is strongly induced in the presence of pectic derivatives, and that the deletion mutant is unable to grow on oligogalacturonides longer than trimers. Such a deletion mutant is also less virulent. Electrophysiological experiments with planar lipid bilayers indicate that KdgM is a voltage-dependent porin, which is slightly selective for anions and exhibits fast current fluctuations in the presence of trigalacturonate.

Homologues of KdgM can be found in other bacteria. NanC is an outer membrane channel protein of *E. coli* allowing the entry of *N*-acetylneuraminic acid (Neu5Ac or sialic acid) into the bacteria. NanC proteins form high-conductance channels which are open at low membrane potentials and which exhibit weak anion selectivity (Condemine et al., 2005). KdgN from *E. chrysanthemi* shows a high sequence identity to KdgM and strong evidence was found that it has the same substrate as KdgM (personal communication G. Condemine).

In the context of membrane transport protein classification, KdgM and NanC are members of the oligogalacturonate-specific porin (KdgM) family belonging to the OmpG superfamily (Blot et al., 2002; Condemine et al., 2005). Interestingly, in contrast to most porins, KdgM was found to be monomeric (Condemine et al., 2005). This coincides with the finding that OmpG, a 14-stranded β-barrel, also seems to be monomeric (Yildiz et al., 2006).

We present here the projection structures of KdgM, KdgN and NanC based on 2D electron crystallogaphy.

Materials and methods

Protein preparation

The three porins were produced without their signal sequence in inclusion bodies followed by a renaturation step. The plasmid pYjh3 used to produce NanC has been described in Condemine et al. (2005). To produce KdgM, an *Nde*I site was inserted into the plasmid pKM2 immediately downstream of the signal peptidase cleavage site (Blot et al., 2002) using the primer Leadd (ctctggttagcgtcaatcatatggctgtatctatcg) and Leadi (gtcgatagatacagccatatgattgacgctaaccag) using the Quickchange Kit (Stratagene). An *Nde*I-*Eco*RI fragment was cut from the plasmid and inserted into the pT7-7 plasmid (Tabor and Richardson, 1985), resulting in plasmid pLead4. To produce KdgN without its signal sequence, the gene was amplified using the primer kdgn10 (gcgggccatatggacattcgactatcgccatgag, *Nde*I site underlined) and kdgn11 (ccgccgaagcttcgggtatcagaagttgtactg, *Hin*dIII site underlined). The amplified fragment was digested with *Nde*I and *Hin*dIII and inserted into the plasmid pET20 (Novagen) and digested with the same enzymes resulting in plasmid pKdgN8.

The plasmids pLead4 and pKdgN8 were introduced into the strain Omp8 (DE3) (Prilipov et al., 1998). The strains were grown in LB medium to an OD600 of 1.0. After 2 h of induction with IPTG, cells were harvested, disrupted in a French cell in 50 mM tris-HCl pH 8.0, 5 mM EDTA. Unbroken cells were eliminated by centrifugation for 5 min at 3,000 g and inclusion bodies were then pelleted by centrifugation for 10 min at 8,000 g. Inclusion bodies were resuspended in 10 mM tris-HCl pH 8.0 at a concentration of 10 mg/ml, and 6 M urea was added. After dissolution of the inclusion bodies, the solution was centrifuged at 20,000 g for 5 min and the supernatant was dialyzed overnight in 10 mM tris-HCl pH 8.0 / 0.5% SDS. The proteins were purified on preparative SDS-PAGE and electroeluted in 25 mM tris / 192 mM glycine / 0.025% SDS.

Sample preparation

All proteins were diluted in the dialysis buffer containing different detergents to obtain the desired final concentration of 1.4 – 1.8 mg/ml. The phospholipids used to prepare the proteoliposomes were dissolved in chloroform at a concentration of 10 mg/ml, dried under a stream of argon, further dried in a dessicator and weighted. Nanopure water was added, and the lipid solutions were then sonicated for 2 min in a cold water bath using a Branson sonifier 250 tip at 20% of the maximum power. The amount of lipid to add to the protein-detergent mix was calculated according to the respective lipid-to protein ratio (LPR).

We used several detergents in different concentrations (DM (0.4%), OG (1%), LDAO (0.25%, 0.4%, 1%), C_8E_4 (0.4%, 0.6%, 1%), Cymal6 (0.4%, 1%), SDS (0.3%), FosCholine9 (1.5%), purchased from Anatrace (Ohio, USA)) and different lipids (DMPC, *E. coli* lipids, DOPC/DMPC, DLPC/DMPC, Cardiolipin and Soybean lipids, all purchased from Avanti Polar Lipids Inc. (Alabaster, USA)) to screen crystallization conditions of KdgM. Additionally, Cholesterol (0.6%, 2.2%) was added to the ternary mixtures for several conditions (see Table 1). To screen conditions for NanC and KdgN we used LDAO (1%) or OG (1%) and different lipids (DMPC, *E. coli* lipids, Cardiolipin, Soybean lipids). The LPRs are indicated in Table 1 and ranged from 0.02 to 0.4.

Crystallization screenings

Initial screenings were performed using the newly established cyclodextrin method described in Signorell et al. (2006). In brief, detergent is removed from a ternary mixture by adding 200µl of a 0.5% cyclodextrin solution to 20µl of ternary mixture over 72 h or 144 h with a dilution device described in Remigy et al. (2003). To have precise starting conditions, the protein was dialyzed overnight against the buffer containing the desired salts and detergent.

Dialysis was performed in 60 µl buttons, sealed with a membrane having a cut-off of 10 kDa. These buttons where then put into a 3 liter flask of dialysis buffer and dialyzed for 3 weeks at room temperature. All the buffers contained nanopure water, a certain amount of salt (see Table 1) and azide. The pH was adjusted to pH 9.0 for tris-HCl and to pH 7.4 for Hepes-KOH.

Electron Microscopy

Negative stain electron microscopy was performed to screen the crystallization attempts. The specimens were prepared by adsorbing proteoliposomes onto carbon film, which was rendered hydrophilic by glow discharging in air. The grids were washed three times with nanopure water and then stained with uranyl acetate (1%). Micrographs were taken on a Hitachi H-7000 transmission electron microscope operated at 100 kV using magnifications of 5’000 to 50’000. An optical bench with a laser beam was used to evaluate the diffraction quality of the micrographs. Samples with well ordered crystals were subsequently utilized for data collection using cryo electron microscopy: The specimens were embedded in 2% glucose on molybdenum grids, which were covered with a carbon film that was previously evaporated onto mica, and floated on water and picked up with the grid. Electron micrographs were taken at low dose (< 10 electrons per $Å^2$) with a Philipps CM-200 FEG microscope operated at 200 kV at a magnification of 50’000.

Image processing

The negatives recorded on Kodak SO163 film were scanned at 2000 lines/cm on a Heidelberg Primescan D 7100, resulting in a resolution of 1 Å/pixel at the sample level. The digitized images were processed using the MRC image processing package (Crowther et al., 1996) and the IPLT image processing toolbox (Philippsen et al., 2006). The epitaxial twinned lattice could be separated in Fourier space using IPLT’s GUI (see below), and the two individual lattices of the upper and lower layer of the flattened tube or vesicle were subsequently processed individually. The selection of an initial spot list as reference and an initial fit of the zero lines in the CTF to determine the defocus were also done manually using IPLT. Overlapping spots belonging to both lattices were automatically determined and excluded from the spot list using the information given by the two lattices. The rest of the image processing was performed using the well established processing scheme of MRC: First an initial reference was calculated from the manually selected spots in the power spectrum of the image, and then two unbending steps were performed. Finally the image was masked to select the crystalline areas. The masked image was processed further, with three unbending steps. The data from the single images was then merged with ORIGTILTK, which does a proper alignment of the single image averages. Refinement of the defocus values, determined earlier by hand, was also activated to minimize the phase error.

The newly developed GUI within IPLT allows to create several overlays having different functions (Philippsen et al., 2006). They are superimposed onto the image (in our case the power spectrum of a scanned image area) and can be displayed, manipulated or locked individually. The current parameters from these overlays are accessible through a Python interface and can be saved for further processing. The overlays utilized for processing were (a) the lattice overlay, allowing the user to fit one or more lattices to the visible spots in the Fourier transform; (b) the CTF overlay, fitting the zero-lines to determine the defocus and astigmatism of an image; and (c) the spot list overlay to select lattice spots from a given lattice.

Calculation of Projection Maps from 3D Datasets

2D projection maps of OmpF and OmpG were calculated based on the atomic coordinates from (Cowan et al., 1995) (PDB accession code 2OMF) and (Yildiz et al., 2006) (PDB accession code 2IWW), using SFALL and FFT from the CCP4 suite (CCP4, 1994). The B-factors were set to 80 and the resolution was set to 4 Å.

Secondary structure prediction

Secondary structure prediction was performed with algorithms specialized in the prediction of transmembrane β-barrels (TMB), namely (a) PROFtmb (Bigelow et al., 2004), based on a profile-based hidden Markov model (HMM) for the prediction and discrimination of TMBs; (b) hmmb2tmr (Martelli et al., 2002), also based on a HMM; and (c) b2tmr (Jacoboni et al., 2001), based on a neural network (NN). All three methods use multiple sequence alignment to include evolutionary information.

Results and Discussion

Crystallization

We tested several hundred 2D crystallization conditions with different lipids and LPRs, different kinds and concentrations of detergents and with buffers at different pH containing different salts. A systematic search in the condition space would be a straightforward brute-force approach to identify good crystallization conditions. However, since automated sample preparation and screening are not available, this would be both time and labor intensive. One bottleneck is the long removal time for low CMC detergents using dialysis. Using cyclodextrin for detergent removal (Signorell et al., 2006), we could shorten the duration of the initial screening considerably. This method also reduces the required amount of purified protein, since it can be applied to much smaller starting volumes than that of dialysis buttons. An initial coarse screening for detergents and lipids with the cyclodextrin method showed preferred crystal formation using LDAO and DMPC. The found crystals were small (100 – 200 nm) but the crystallization conditions could be refined in a larger screening of conditions using dialysis buttons.

The nature of the used lipid was of great importance in the formation of proteoliposomes: Samples with several lipids (DMPC, *E. coli* lipids, DOPC/DMPC, DLPC/DMPC, cardiolipin and soybean lipids) were tested, but only the samples prepared with DMPC showed an insertion of protein into the lipid vesicles.

The LPR was as critical for crystal formation as the nature of the lipid affected chosen. An LPR of

0.15, for example, yielded crystals of KdgM whereas LPRs of 0.1 and 0.2 did not result in crystal formation. Such sensitivity to the LPR could be observed for all three proteins. Table 1 summarizes the dialysis experiments with DMPC. It gives a wider range of LPRs yielding crystals. However this table summarizes experiments from several protein batches, each of which had its protein concentration determined separately introducing a small uncertainty in the protein concentration. Both the presence of aggregates in the samples and the narrow range of LPRs yielding crystals, suggests that crystal formation requires a certain packing density that is very close to a state of disordered aggregation. This finding is also supported by the low LPR where the best crystals for each of the proteins were found: KdgM and NanC at LPR 0.1-0.15, KdgN at LPR 0.2.

The choice of detergent turned out to be a critical parameter as well. Of all the tested detergents (see materials and methods) only few facilitated crystal formation. KdgM crystals were found using LDAO, OG and C_8E_4, indicating a wider acceptance of detergents. KdgN crystals were found using OG and LDAO, while NanC only yielded crystals using LDAO, a detergent with a relatively low CMC. The best quality of crystals in terms of size and order could be found using 1% LDAO for KdgM and NanC and 1% OG for KdgN. The concentration of the detergents had only a minor influence in the range tested.

Most of the experiments were conducted at pH 9 since initial experiments with KdgM have shown that crystallization occurred most often at this pH. All three proteins crystallized at pH 9, but since the result of KdgN was not satisfying, experiments at pH 7.4 in combination with different salts (see below) were performed. The best KdgN crystals were found at pH 7.4, whereas pH 9 yielded better results for KdgM and NanC.

Salt addition to the dialysis buffer had a significant influence on the formation and the quality of the crystals. The salt compositions used in the different dialysis experiments are shown in Table 1. The use of $MgCl_2$ resulted in the best NanC crystals, while preventing crystal formation for KdgM. NaCl was mandatory for the formation of crystals when no other monovalent salt was used (KCl experiments), but the formation of crystals was less dependent on the concentration. For KdgM, where the widest screen of crystallization conditions was performed, crystals assembled at NaCl concentrations between 100 mM and 500 mM NaCl (see Table 1).

The best KdgM crystals were obtained using 1% LDAO at an LPR of 0.1 and a buffer containing 200 mM NaCl at pH 9. The dialysis experiments yielded crystalline tubes with a length of up to 1 µm (see Fig. 1A). Adding cholesterol with the goal to increase the size of tubes or to induce sheet formation did not yield the desired result but neither did it prevent the formation of crystalline tubes.

Well ordered tubular crystals up to 1 µm in length were the best results for NanC (see Fig 1B). 1% LDAO, an LPR of 0.1 and a buffer at pH 9 containing 1 mM $MgCl_2$ and 200 mM NaCl were used.

KdgN did not crystallize as well as the two other proteins. In contrast to KdgM and NanC, crystalline tubes and crystalline vesicles were found, the latter exhibiting better ordering. The conditions that yielded the best crystals were 1% OG, DMPC with an LPR of 0.2 dialyzed in a buffer of pH 7.4 containing 100 mM KCl. The size of the vesicles (see Fig 1C) was only slightly smaller (about 700 to 800 nm) than the size of the crystalline KdgM and NanC tubes. The vesicles tended to flatten less well on the carbon film than the tubes and the edges of the vesicle then consisted of stacked multiple layers. Since this stacking resulted in additional spots in the power spectrum, which could not be assigned to one of the lattices, only part of the crystalline area could be used for image processing.

Even if KdgM and KdgN are very similar in terms of sequence and packing arrangement of the monomers (see below), the conditions where crystals formed were quite different. This is somewhat unexpected, since the more distant (in terms of sequence and monomer arrangement in the crystal) NanC had similar crystallization conditions as KdgM.

The crystals of all three proteins were well ordered and allowed the determination of projection maps containing information to a resolution beyond 7 Å (see supplementary information SI 1).

Image Processing and Projection Map Analysis

The collapsed tubes and vesicles yielded double layered crystals, imposing the well-known problem of separating information from two different crystalline layers. The diffraction spots of the two lattices could be separated for all crystal types and the layers were processed individually after masking. This was facilitated by IPLT's GUI (Philippsen et al., 2006), which provides the possibility to display and manipulate the two lattices simultaneously. This feature was particularly useful in the case of KdgM, where the lattices from the upper and lower crystalline layers are rotated by approximately 90° with respect to each other making it difficult to assign a spot to one of the two lattices. This is exemplified in Fig. 2A, where spots from both lattices are very close together. Since the inner orders in the power spectrum were weak, one could misleadingly assume a square lattice.

A similar problem was observed with the images of NanC. Here, the two lattices are rotated only slightly with respect to one another, but in this case the unit cell is square. Spots from the two lattices are thus very close to each other, even for the higher reflections. The lattice assignment was greatly facilitated by the simultaneous display of the two lattices (see Fig. 2B).

The unit cell vectors of KdgM were determined to be a=40.4±1.1Å, b=115.4±1.1Å and γ=90±0.8° (n=18), with 4 monomers per unit cell. Exploring different packing arrangements by ALLSPACE of the MRC suite (Crowther et al., 1996), the best symmetry was found to be $p2_12_12$, meaning that pairs of monomers are packed in an upside down manner (see Fig 3A). Each monomer has the shape of an elliptical pore (see Fig. 4A) with the major axis having a length of 24 Å and the minor axis 20 Å giving a ratio of 1.2.

The unit cell vectors of KdgN are a=42.0±1.0Å, b=115.7±1.3Å and γ=90±1.7° (n=6), almost identical to the unit cell vectors of KdgM. The packing arrangements deduced with ALLSPACE again proposed $p2_12_12$ to be the best symmetry (see Fig. 3A). The monomer in the crystal (see Fig. 4C) has dimensions of 23 Å for the long axis and 20 Å for the short axis, giving a ratio of 1.15.

The unit cell vectors of NanC are: a=b=123.6±1.1Å, and γ=90±0.4° (n=8), leading to a unit cell consisting of 12 monomers, the best fitting symmetry also being $p2_12_12$. Unlike KdgM and KdgN, NanC has a threefold pseudo-symmetry, resulting in three slightly different monomers per asymmetric unit (see Fig. 3B). Each monomer shows a shape of an elliptical pore (see Fig. 4B) with dimensions of 24 Å for the long axis and 17 Å for the short axis, giving an axial ratio of 1.4.

From the presented measurements it can be seen that the size of the protein pores and their packing density is very similar for all three proteins. The higher LPR in the crystallization conditions for KdgN as opposed to those of KdgM and NanC can be explained with the observation of empty lipid vesicles in addition to crystalline vesicles in the KdgN samples.

To compare these results with the known structures of the porins OmpF (Cowan et al., 1995) and OmpG (Yildiz et al., 2006), we calculated density projection maps perpendicular to the lipid plane using the atomic coordinates of the according 3D dataset (see Fig. 4). The single OmpF pore (see Fig. 4D) has dimensions of 33 Å for the major axis and 28 Å for the minor axis, leading to a ratio of 1.2 between the axes. The single OmpG pore (see Fig. 4E) has dimensions of 26 Å for the major axis and 18 Å for the minor axis, giving an axial ratio of 1.4.

These sizes and ratios of the pores are in the same range as those of KdgM, NanC and KdgN. The positions of the pores (lower density in the projections) of KdgM, NanC and KdgN in the monomer show more similarity to the position of the pore in OmpG than in the bigger OmpF monomer, where a low density region of roughly the same size is clearly on one side of the monomer (see Fig. 4) .This suggests that KdgM, NanC and KdgN are structurally more similar to OmpG than to OmpF, compatible with the fact that the KdgM family members the OmpG superfamily (Blot et al., 2002; Condemine et al., 2005). The projections of the monomers show some degree of asymmetry in the β-barrel. This is more outstanding in OmpF and less visible in KdgM (see Fig. 4A and E).

The single monomers forming the OmpF trimer are in direct contact, opposing the observed arrangement in the presented porins, where the spacing of all monomers is regular. This is particularly worth mentioning for the triplets of NanC pores forming the asymmetric unit, where the monomers composing one triplet are not closer together than the monomers not belonging to the same triplet (see Fig. 3).

Secondary Structure Prediction and Determination of Periodicity

By comparison of the projection maps of KdgM, KdgN and NanC with those calculated from the structures of OmpF and OmpG, we can assume that the former proteins all form β-barrels. Methods specialized in the secondary structure prediction of membrane spanning proteins have been used to predict the location of the β-strands. In support of our assumption, the algorithm described in Bigelow et al. (2004) predicted all three proteins to be transmembrane β-barrels. The number of predicted transmembrane strands with this method and also with the method described in Martelli et al. (2002) was 12 for all three proteins (see Fig. 5). The NN method described in Jacoboni et al. (2001) predicted 10 transmembrane strands for KdgM and KdgM and 12 transmembrane strands for NanC. This NN prediction shows nevertheless a very similar overall distribution of β-strands to the predictions from Martelli et al. (2002) and Bigelow et al. (2004) (see Fig. 5). The missing (KdgM and KdgN) or differing (NanC) strands in the prediction from Jacoboni et al. (2001) are likely to be absent due to a wrong prediction, since the presence of such long internal loops does not seem very likely. The predictions of all three methods result in overall longer external loop segments compared to the predictions presented in Pellinen et al. (2003), where 14 β-strands were predicted, nevertheless with external loops longer than periplasmic ones. Loop 5 in the HMM predictions fits the main restriction site (loop 6) stated in Pellinen et al. (2003) in terms of sequence. The longer external loops in the HMM predictions are furthermore in agreement with the structure of the 14 stranded OmpG porin (Yildiz et al., 2006) where 7 longer loops are present on the extracellular side. The claimed prediction accuracy of the algorithms from Bigelow et al. (2004) and Martelli et al. (2002) and the fact that their prediction for OmpG was also correct favors the HMM models of 12 transmembrane β-strands in all three proteins.

Conclusions and Outlook

The limited size of the crystals and their tubular shape made it difficult to collect images of tilted samples of a suitable quality to calculate a 3D map of the porins. The processing of tilted KdgM crystals (data not shown) showed a necessity for larger sheet-like crystals, since the usable area of the crystalline tubes was very limited in one dimension. A single layer crystal would solve the problem of the assignment of spots to the correct lattice, which proved to be difficult for images of untilted samples and almost impossible for the images of tilted samples. The performed screening for crystallization conditions was extensive, yet not the whole condition space could be sampled. The search for conditions could be extended even more in order to produce larger or single layered crystals. To master this high number of conditions a more automated procedure concerning sample preparation and screening at the microscope would be required.

The calculation of the projection maps of the three proteins we crystallized show β-barrel structures similar to the structure of OmpG and OmpF. Similar to OmpG, of which 2D crystals were produced as well (Yildiz et al., 2006), all three proteins studied here are monomeric in the crystal and they all show $p2_12_12$ symmetry. The secondary structure prediction performed suggests a β-barrel structure containing 12 β-strands which would be among the smallest number of strands observed in porins. This hypothesis can only be proven by a 3D map of these porins, which in turn requires crystals of larger size and better quality.

Acknowledgments

This work was supported by the Maurice E. Müller Foundation of Switzerland, by the Swiss National Foundation (SNF) within the framework of the National Center of Competence in Research for Structural Biology, the SNF grant 3100-059415 to AE, the NoE 3D-EM (EU project LSHG-CT-2004-502828), and the HT-3DEM (EU project LSHG-CT-2005-018811).

References

Bigelow, H.R., Petrey, D.S., Liu, J., Przybylski, D., and Rost, B., 2004. Predicting transmembrane beta-barrels in proteomes. Nucleic Acids Res, 32(8):2566-77.

Blot N., Berrier, C., Hugouvieux-Cotte-Pattat, N., Ghazi, A. and Condemine, G., 2002. The oligogalacturonate-specific porin KdgM of *Erwinia chrysanthemi* belongs to a new porin family. J Biol Chem, 277(10):7936-44.

Condemine, G., Berrier, C., Plumbridge J., and Ghazi, A., 2005. Function and expression of an N-acetylneuraminic acid-inducible outer membrane channel in *Escherichia coli*. J Bacteriol, 187(6):1959-65.

Cowan, S.W., Garavito, R.M., Jansonius, J.N., Jenkins, J.A., Karlsson, R., Konig, N., Pai, E.F., Pauptit, R.A., Rizkallah, P.J., Rosenbusch, J.P., 1995. The structure of OmpF porin in a tetragonal crystal form.. Structure, 3(10):1041-50.

Crowther, R.A., Henderson R. and Smith, J.M., 1996. MRC image processing programs. J Struct Biol, 116(1):9-16.

Jacoboni, I., Martelli, P.L., Fariselli, P., De Pinto, V. and Casadio, R., 2001. Prediction of the transmembrane regions of beta-barrel membrane proteins with a neural network-based predictor. Protein Sci, 10(4):779-87.

Koebnik, R., Locher, K.P. and Van Gelder, P., 2000. Structure and function of bacterial outer membrane proteins: barrels in a nutshell. Mol Microbiol, 37(2):239-53.

Martelli, P.L., Fariselli, P., Krogh, A. and Casadio, R., 2002. A sequence-profile-based HMM for predicting and discriminating beta barrel membrane proteins. Bioinformatics, 18 Suppl 1:S46-53.

Pellinen, T., Ahlfors, H., Blot, N. and Condemine G., 2003. Topology of the *Erwinia chrysanthemi* oligogalacturonate porin KdgM. Biochem J, 372(Pt 2):329-34.

Philippsen, A., Schenk, A.D., Signorell, G.A., Mariani, V., Berneche S. and Engel, A., 2006. Collaborative EM image processing with the IPLT image processing library and toolbox. J Struct Biol 157: 28-37.

Prilipov, A., Phale, P.S., Van Gelder, P., Rosenbusch, J.P. and Koebnik, R., 1998. Coupling site-directed mutagenesis with high-level expression: large scale production of mutant porins from E. coli. FEMS Microbiol Lett, 163(1):65-72.

Remigy, H.W., Caujolle-Bert, D., Suda, K., Schenk, A., Chami, M. and Engel, A., 2003. Membrane protein reconstitution and crystallization by controlled dilution. FEBS Lett, 555(1):160-9.

Schirmer, T, 1998. General and specific porins from bacterial outer membranes.. J Struct Biol, 121(2):101-9.

Schulz, G.E., 1996. Porins: general to specific, native to engineered passive pores. Curr Opin Struct Biol, 6(4):485-90.

Signorell, G.A., Kaufmann, T.C., Kukulski, W., Engel, A. And Remigy, H.W., 2006. Controlled 2D crystallization of membrane proteins using methyl-beta-cyclodextrin. J Struct Biol 157:321-328.

Tabor, S. and Richardson, C.C., 1985. A bacteriophage T7 RNA polymerase/promoter system for controlled exclusive expression of specific genes. Proc Natl Acad Sci U S A, 82(4):1074-8.

Yildiz, O., Vinothkumar, K.R., Goswami, P. and Kuhlbrandt, W., 2006. Structure of the monomeric outer-membrane porin OmpG in the open and closed conformation. EMBO J, 25(21):5240.

CCP4, 1994. The CCP4 suite: programs for protein crystallography. Acta Crystallogr D Biol Crystallogr, 50(Pt 5):760-3.

Figures and Tables

Figure 1

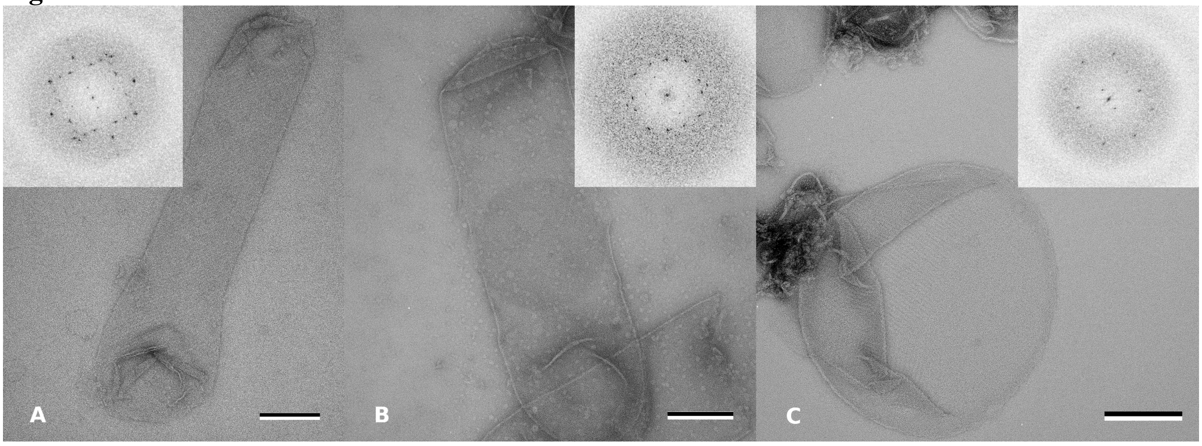


Figure 1: Negative stain images of protein crystals and their power spectrums. The scale bars on all images indicate 200 nm. A) KdgM tube showing a well ordered crystal. This tube is rather long and thin, shorter and wider tubes have also been observed. B) Crystalline NanC tubes. The 'stripes' from the up-down arrangement of the two layers form a checker. C) Vesicle of crystalline arranged KdgN. The flat and double layered area is quite small compared to the overall size of the vesicle due to the folding at the border of the vesicle.

Figure 2

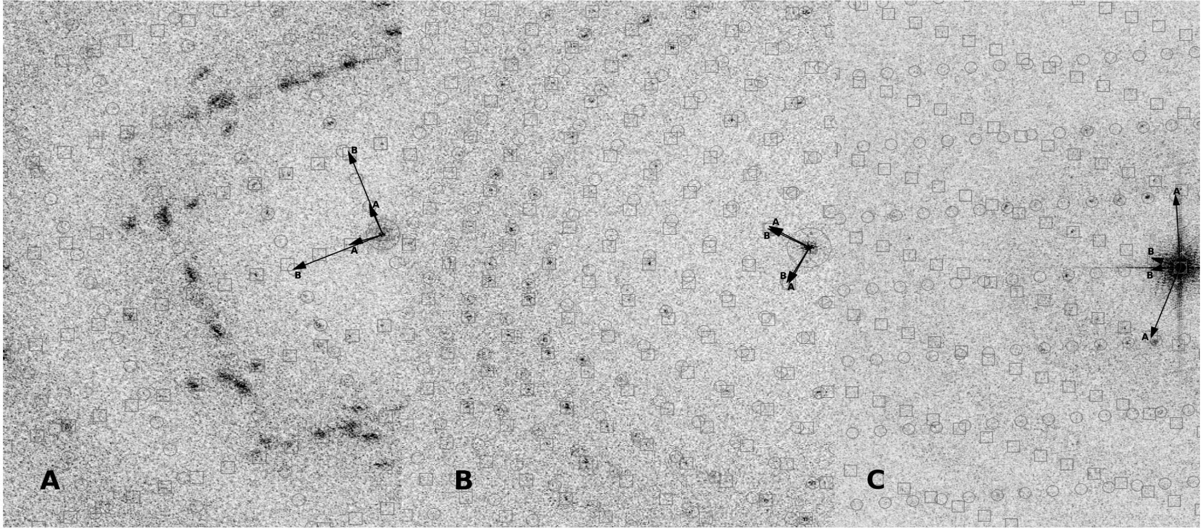


Figure 2: Calculated power spectra of cryo electron micrograps of KdgM, NanC and KdgN with both lattices indexed. The two lattices are displayed with the IPLT GUI. A) Power spectrum of a KdgM crystal. The relative arrangement of the two lattices (rotated by ca. 90° in respect to each other) and the missing inner reflections make it difficult to distinguish spots belonging to the top (circles) or the bottom (squares) layer of the crystal tube. B) Power spectrum of a NanC crystal tube. The two lattices are very close one to another, since the lattices are square and the rotation of the two lattices in respect to eachother equals 90°. Additianaly not all reflections are strong, thus the discrimination of the top (circles) and the bottom (squares) layers is difficult. C) Power spectrum of a crystalline KdgN vesicle. The single lattices resemble those of KdgM but the rotaion between the two lattices is not constant.

Figure 3

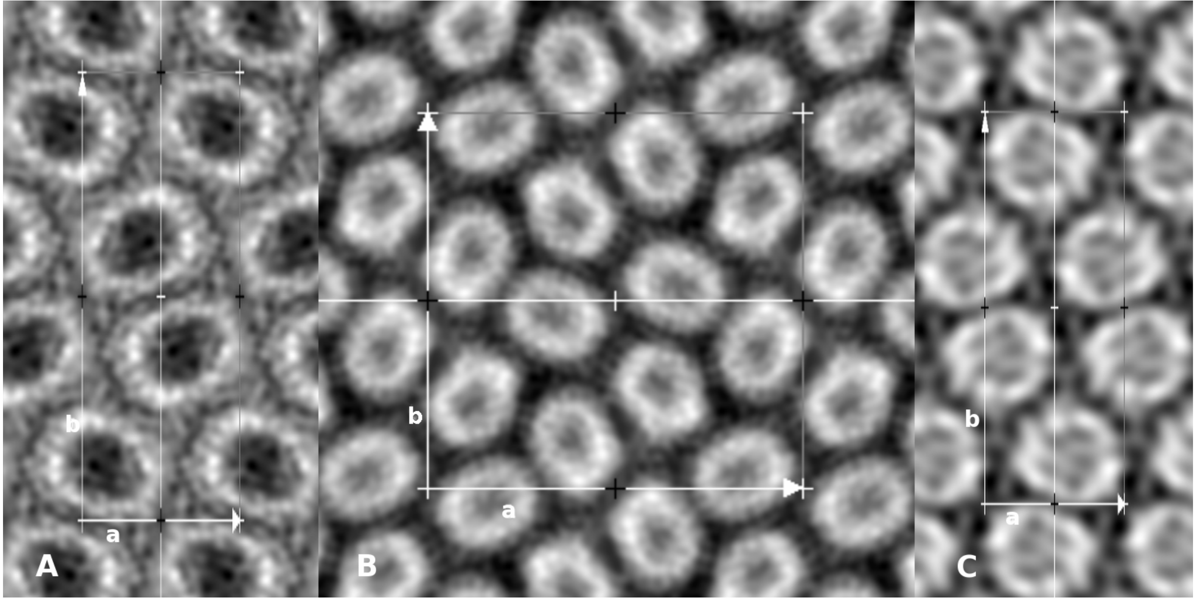


Figure 3: Calculated projection maps of the three porins. A) Projection map of KdgM. The image has a resolution of 5 Å and is symmetrized using p$2_1 2_1$2 symmetry. The unit cell vectors are a = 115 Å and b = 40 Å. B) Projection map of NanC. The image has a resolution of 5 Å and is symmetrized using p$2_1 2_1$2 symmetry. The unit cell vectors are a = b = 124 Å. The monomers are arranged in a 3 fold non crystallographic symmetry. One asymmetric unit consist of one of these triplets. C) Projection map of KdgN. The image has a resolution of 7 Å and is symmetrized using p$2_1 2_1$2 symmetry. The unit cell vectors are a = 115 Å and b = 41 Å.

Figure 4

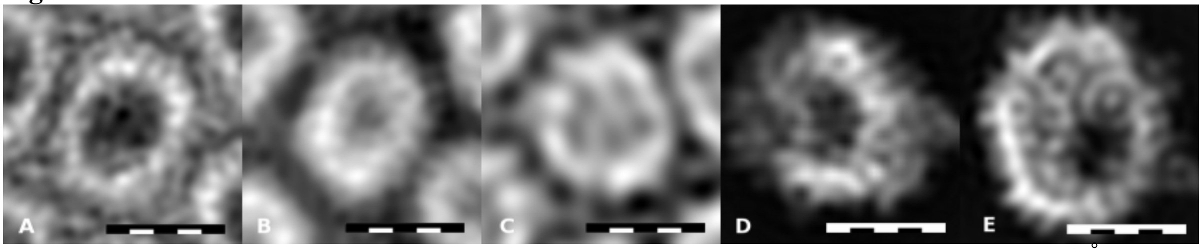
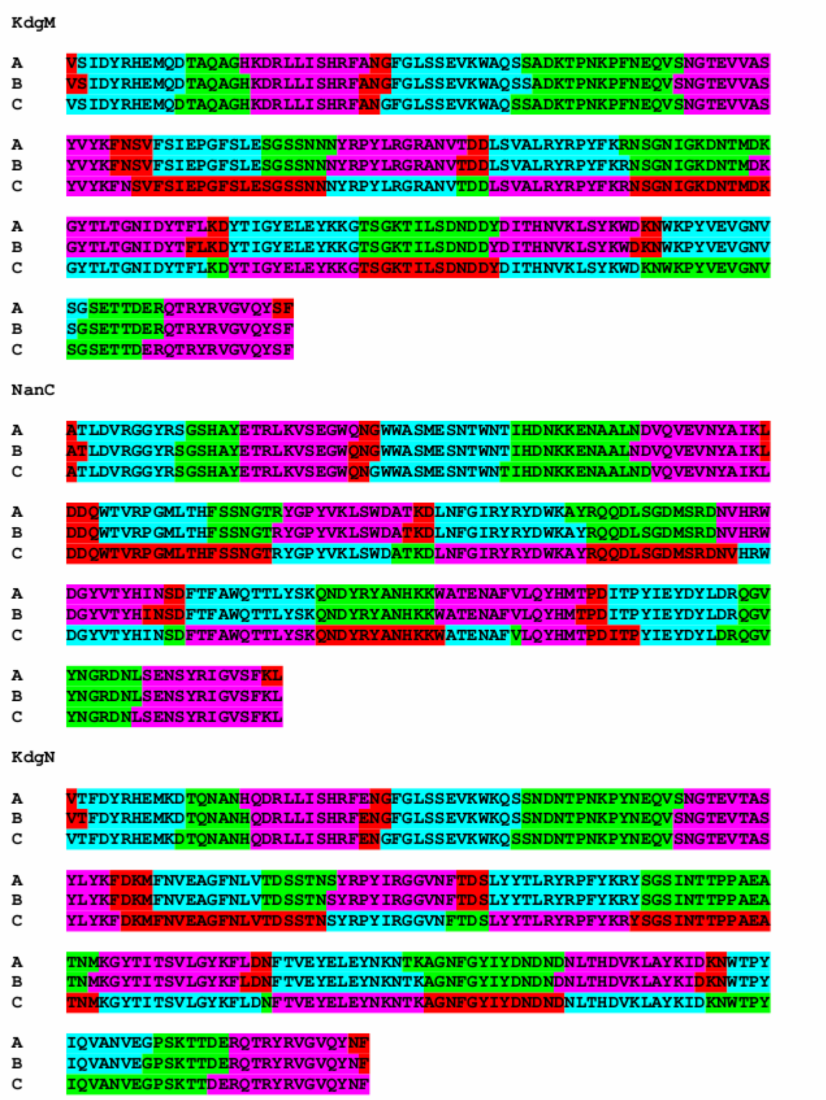


Figure 4: Monomers of KdgM, NanC, KdgN, OmpG and OmpF. The scale bars represent 25 Å. A)-C): KdgM (A), NanC (B) and KdgN (C) monomers calculated from cryo-EM images. KdgM and NanC have a resolution of 5 Å and KdgN a resolution of 7 Å. D): OmpG monomer calculated from the PDB file 2IWW at a resolution of 4 Å. E): OmpF monomer calculated from the PDB file 2OMF at a resolution of 4 Å. The pores (low density) of KdgM, NanC and KdgN are in the center of the monomers, what can also be seen in the calculated projection of OmpG, which accommodates 14 β-strands. OmpF has a pore of similar size on one side of the monomer, which is built from 16 β-strands.

Figure 5

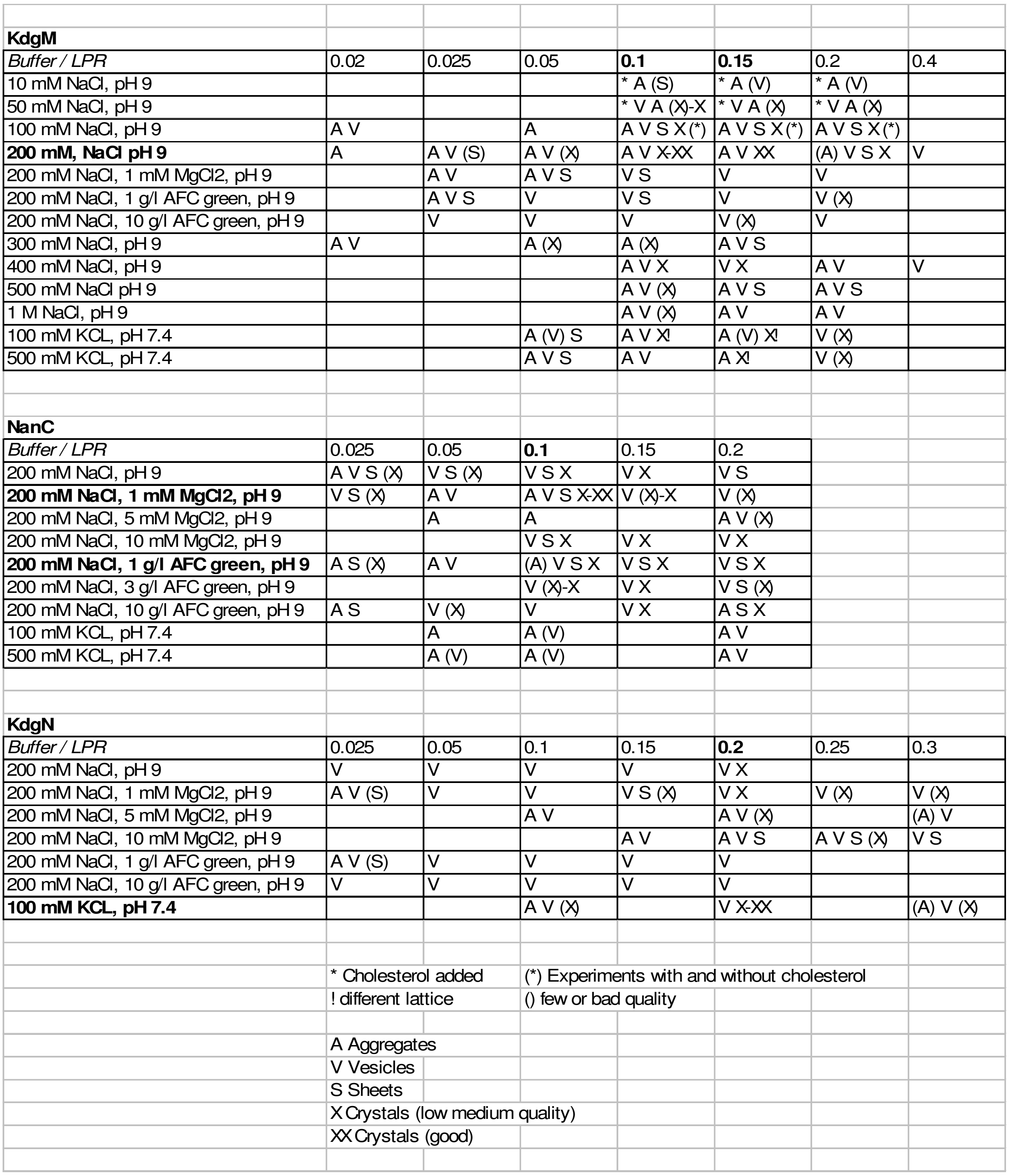


```
KdgM

A     VSIDYRHEMQDTAQAGHKDRLLISHRFANGFGLSSEVKWAQSSADKTPNKPFNEQVSNGTEVVAS
B     VSIDYRHEMQDTAQAGHKDRLLISHRFANGFGLSSEVKWAQSSADKTPNKPFNEQVSNGTEVVAS
C     VSIDYRHEMQDTAQAGHKDRLLISHRFANGFGLSSEVKWAQSSADKTPNKPFNEQVSNGTEVVAS

A     YVYKFNSVFSIEPGFSLESGSSNNNYRPYLRGRANVTDDLSVALRYRPYFKRNSGNIGKDNTMDK
B     YVYKFNSVFSIEPGFSLESGSSNNNYRPYLRGRANVTDDLSVALRYRPYFKRNSGNIGKDNTMDK
C     YVYKFNSVFSIEPGFSLESGSSNNNYRPYLRGRANVTDDLSVALRYRPYFKRNSGNIGKDNTMDK

A     GYTLTGNIDYTFLKDYTIGYELEYKKGTSGKTILSDNDDYDITHNVKLSYKWDKNWKPYVEVGNV
B     GYTLTGNIDYTFLKDYTIGYELEYKKGTSGKTILSDNDDYDITHNVKLSYKWDKNWKPYVEVGNV
C     GYTLTGNIDYTFLKDYTIGYELEYKKGTSGKTILSDNDDYDITHNVKLSYKWDKNWKPYVEVGNV

A     SGSETTDERQTRYRVGVQYSF
B     SGSETTDERQTRYRVGVQYSF
C     SGSETTDERQTRYRVGVQYSF

NanC

A     ATLDVRGGYRSGSHAYETRLKVSEGWQNGWWASMESNTWNTIHDNKKENAALNDVQVEVNYAIKL
B     ATLDVRGGYRSGSHAYETRLKVSEGWQNGWWASMESNTWNTIHDNKKENAALNDVQVEVNYAIKL
C     ATLDVRGGYRSGSHAYETRLKVSEGWQNGWWASMESNTWNTIHDNKKENAALNDVQVEVNYAIKL

A     DDQWTVRPGMLTHFSSNGTRYGPYVKLSWDATKDLNFGIRYRYDWKAYRQQDLSGDMSRDNVHRW
B     DDQWTVRPGMLTHFSSNGTRYGPYVKLSWDATKDLNFGIRYRYDWKAYRQQDLSGDMSRDNVHRW
C     DDQWTVRPGMLTHFSSNGTRYGPYVKLSWDATKDLNFGIRYRYDWKAYRQQDLSGDMSRDNVHRW

A     DGYVTYHINSDFTFAWQTTLYSKQNDYRYANHKKWATENAFVLQYHMTPDITPYIEYDYLDRQGV
B     DGYVTYHINSDFTFAWQTTLYSKQNDYRYANHKKWATENAFVLQYHMTPDITPYIEYDYLDRQGV
C     DGYVTYHINSDFTFAWQTTLYSKQNDYRYANHKKWATENAFVLQYHMTPDITPYIEYDYLDRQGV

A     YNGRDNLSENSYRIGVSFKL
B     YNGRDNLSENSYRIGVSFKL
C     YNGRDNLSENSYRIGVSFKL

KdgN

A     VTFDYRHEMKDTQNANHQDRLLISHRFENGFGLSSEVKWKQSSNDNTPNKPYNEQVSNGTEVTAS
B     VTFDYRHEMKDTQNANHQDRLLISHRFENGFGLSSEVKWKQSSNDNTPNKPYNEQVSNGTEVTAS
C     VTFDYRHEMKDTQNANHQDRLLISHRFENGFGLSSEVKWKQSSNDNTPNKPYNEQVSNGTEVTAS

A     YLYKFDKMFNVEAGFNLVTDSSTNSYRPYIRGGVNFTDSLYYTLRYRPFYKRYSGSINTTPPAEA
B     YLYKFDKMFNVEAGFNLVTDSSTNSYRPYIRGGVNFTDSLYYTLRYRPFYKRYSGSINTTPPAEA
C     YLYKFDKMFNVEAGFNLVTDSSTNSYRPYIRGGVNFTDSLYYTLRYRPFYKRYSGSINTTPPAEA

A     TNMKGYTITSVLGYKFLDNFTVEYELEYNKNTKAGNFGYIYDNDNDNLTHDVKLAYKIDKNWTPY
B     TNMKGYTITSVLGYKFLDNFTVEYELEYNKNTKAGNFGYIYDNDNDNLTHDVKLAYKIDKNWTPY
C     TNMKGYTITSVLGYKFLDNFTVEYELEYNKNTKAGNFGYIYDNDNDNLTHDVKLAYKIDKNWTPY

A     IQVANVEGPSKTTDERQTRYRVGVQYNF
B     IQVANVEGPSKTTDERQTRYRVGVQYNF
C     IQVANVEGPSKTTDERQTRYRVGVQYNF
```

Figure 5: Secondary structure predictions of KdgM, NanC and KdgN resulting from the algorithms presented in Bigelow et al. (2004) (row A), Martelli et al. (2002) (row B) and Jacoboni et al. (2001) (row C). The secondary structure is color coded as follows: red: periplasmic loop or terminus; green: extracellular loop; blue: β-strand periplasm to extracellular medium; magenta: β-strand extracellular medium to periplasm. The HMM predictions (top and middle row) show a different β-strand pattern than the NN prediction (bottom row).

12

Table 1



KdgM

Buffer / LPR	0.02	0.025	0.05	**0.1**	**0.15**	0.2	0.4
10 mM NaCl, pH 9				* A (S)	* A (V)	* A (V)	
50 mM NaCl, pH 9				* V A (X)-X	* V A (X)	* V A (X)	
100 mM NaCl, pH 9	A V		A	A V S X (*)	A V S X (*)	A V S X (*)	
200 mM, NaCl pH 9	A	A V (S)	A V (X)	A V X-XX	A V XX	(A) V S X	V
200 mM NaCl, 1 mM MgCl2, pH 9		A V	A V S	V S	V	V	
200 mM NaCl, 1 g/l AFC green, pH 9		A V S	V	V S	V	V (X)	
200 mM NaCl, 10 g/l AFC green, pH 9		V	V	V	V (X)	V	
300 mM NaCl, pH 9	A V		A (X)	A (X)	A V S		
400 mM NaCl, pH 9				A V X	V X	A V	V
500 mM NaCl pH 9				A V (X)	A V S	A V S	
1 M NaCl, pH 9				A V (X)	A V	A V	
100 mM KCL, pH 7.4			A (V) S	A V X!	A (V) X!	V (X)	
500 mM KCL, pH 7.4			A V S	A V	A X!	V (X)	

NanC

Buffer / LPR	0.025	0.05	**0.1**	0.15	0.2
200 mM NaCl, pH 9	A V S (X)	V S (X)	V S X	V X	V S
200 mM NaCl, 1 mM MgCl2, pH 9	V S (X)	A V	A V S X-XX	V (X)-X	V (X)
200 mM NaCl, 5 mM MgCl2, pH 9		A	A		A V (X)
200 mM NaCl, 10 mM MgCl2, pH 9			V S X	V X	V X
200 mM NaCl, 1 g/l AFC green, pH 9	A S (X)	A V	(A) V S X	V S X	V S X
200 mM NaCl, 3 g/l AFC green, pH 9			V (X)-X	V X	V S (X)
200 mM NaCl, 10 g/l AFC green, pH 9	A S	V (X)	V	V X	A S X
100 mM KCL, pH 7.4		A	A (V)		A V
500 mM KCL, pH 7.4		A (V)	A (V)		A V

KdgN

Buffer / LPR	0.025	0.05	0.1	0.15	**0.2**	0.25	0.3
200 mM NaCl, pH 9	V	V	V	V	V X		
200 mM NaCl, 1 mM MgCl2, pH 9	A V (S)	V	V	V S (X)	V X	V (X)	V (X)
200 mM NaCl, 5 mM MgCl2, pH 9			A V		A V (X)		(A) V
200 mM NaCl, 10 mM MgCl2, pH 9				A V	A V S	A V S (X)	V S
200 mM NaCl, 1 g/l AFC green, pH 9	A V (S)	V	V	V	V		
200 mM NaCl, 10 g/l AFC green, pH 9	V	V	V	V	V		
100 mM KCL, pH 7.4			A V (X)		V X-XX		(A) V (X)

* Cholesterol added (*) Experiments with and without cholesterol
! different lattice () few or bad quality

A Aggregates
V Vesicles
S Sheets
X Crystals (low medium quality)
XX Crystals (good)

Table 1: Summary of tested crystallization conditions using DMPC and LDAO, using dialysis for detergent removal. The table shows an overview of the results of KdgM, NanC and KdgN, varying salts and LPR.

Supplementary Information

S1: Phase residual table of KdgM, NanC and KdgN, calculated with MRC:

			KdgM				NanC				KdgN		
DMIN	DMAX		all IQ's	IQ-weighte	# of reflexes		all IQ's	IQ-weighte	# of reflexes		all IQ's	IQ-weighte	# of reflexes
100	9.7		29.7	26.5	720		32.8	30.7	820		36.2	34.3	146
9.7	6.9		50.1	45.7	238		67.4	63.3	127		51.9	51.4	20
6.9	5.6		92.6	92.6	95		88.5	88.2	74		164.4	163.7	4
5.6	4.9		77.8	77.8	87		69.5	68.5	47		95.5	98.0	9
4.9	4.3		78.9	80.0	61		92.8	92.1	45		64.1	64.1	4
4.3	4.0		88.4	86.8	66		89.3	89.1	60		68.3	65.8	9
4.0	3.7		102.1	103.1	47		110.8	111.1	48		97.2	93.0	8
3.7	3.4		83.3	83.6	59		93.0	94.2	75		97.1	98.1	9
3.4	3.2		86.0	86.2	61		92.3	91.8	80		91.5	94.6	6
3.2	3.1		75.1	74.6	61		86.7	87.4	33		84.2	87.8	4
3.1	3.0		71.9	71.8	23		74.1	72.9	29		52.3	52.3	4
		Overall		53.155	1518			55.273	1438			51.414	223

2.3.7 Additional Data and Outlook

2.3.7.1 AFM Images

The surface topology and the crystalline arrangement of KdgM were investigated using the atomic force microscope (AFM). The intact KdgM tubes were observed to be not completely flat as can be seen in figure 2.17. These closed KdgM tubes could be opened by applying some force with the AFM tip to the surface. This resulted in flat sheet structures (see figure 2.18) suitable for AFM imaging at a high magnification.

The stripes, already visible in both overview and detail images, resulted from the arrangement of the monomers in the crystals. Two up-oriented monomers (bright stripes) are followed by two down-oriented monomers (dark stripes). At higher resolution (see figure 2.19) this arrangement is clearly visible.

At the border of the flat open tubes, KdgM monomers started to disassociate from the crystal (figure 2.19, right border). This is a result of the applied force and the opening of the tube using the AFM tip as a scratching tool.

The AFM images helped us to confirm the crystalline arrangement of the monomers (see below). The fact that the intact tubes are not completely flat also explains, to a certain extent, the limited resolution of the calculated projection maps.

2.3.7.2 Symmetry of the Crystals

To determine the crystal symmetry, the MRC program ALLSPACE was used for each of the calculated projection maps. The symmetry used for the calculation of the projection maps presented in section 2.3.6 was $p22_12_1$ for all three porins. The output of the ALLSPACE program shows, which of the symmetries are the most probable, based on the phase error.

KdgM The crystals of KdgM showed a clear $p22_12_1$symmetry for all processed images. In the figure 2.20, you can see an example of such a statistic. The good values of other symmetries such as $p2$, $p12_1a$ and $p12_1b$ result from the fact, that these symmetries are sub symmetries of $p22_12_1$.

KdgN The lower quality of the KdgN crystals resulted in a less good data quality. This reduced quality explains the less clear results for the symmetry calculations of the KdgN crystals. Figure 2.21 exemplifies this behaviour. By looking at the crystals, we could exclude most of the symmetries having good values. The $p22_12_1$symmetry was chosen, since its phase residual errors were low in most of the processed images and its sub symmetries also had good values.

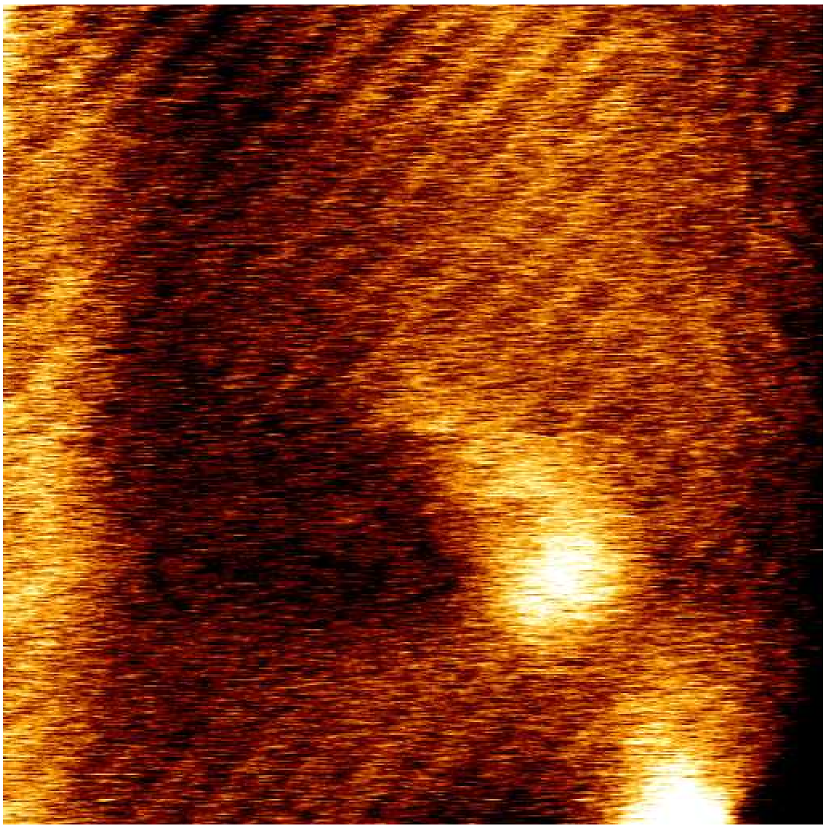


Figure 2.17: AFM Image of KdgM

This image shows the surface of a KdgM tube. The brighter areas show higher regions, the darker areas lower regions. The diagonal stripes result from the two up, two down arrangement of the protein monomers.

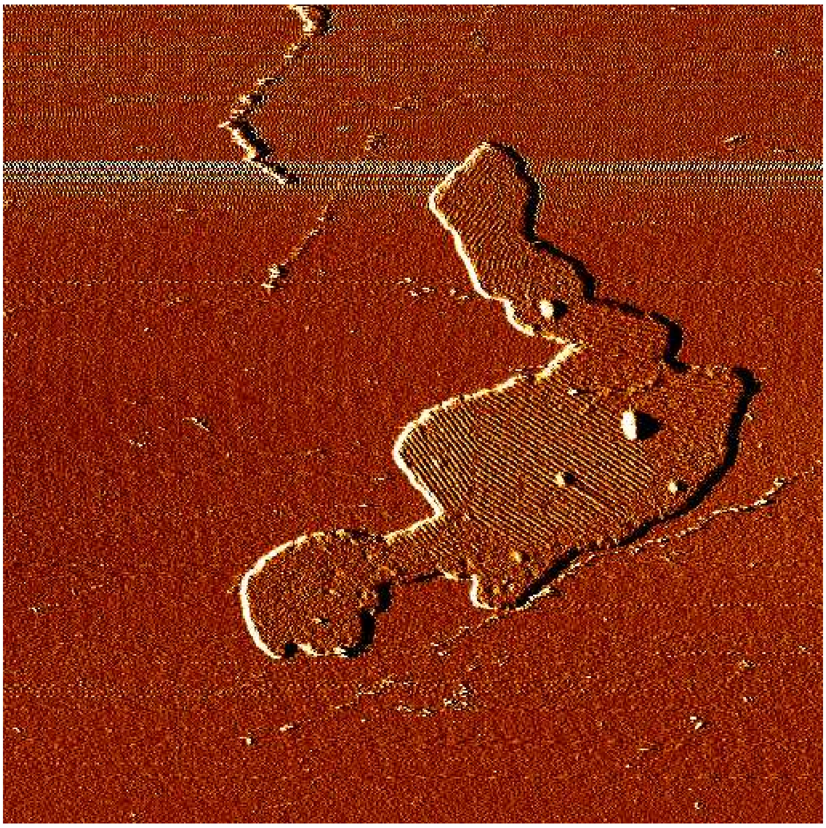


Figure 2.18: AFM Overview of an Opened KdgM Tube
The crystalline tubes can be opened by applying moderate force from the AFM tip to the closed tube. This can result in flat crystalline areas.

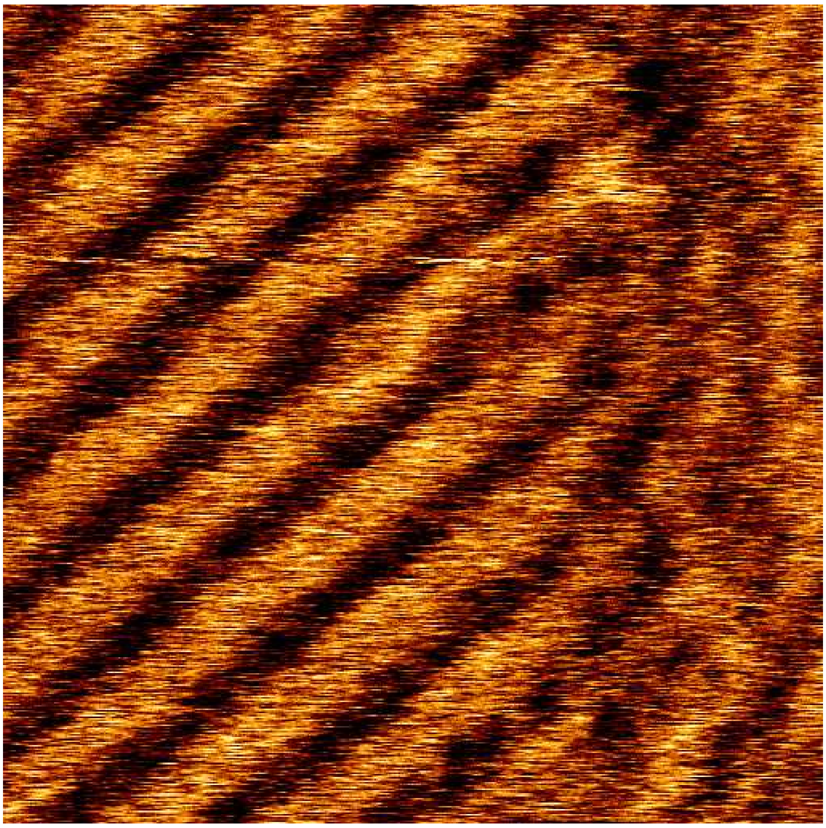
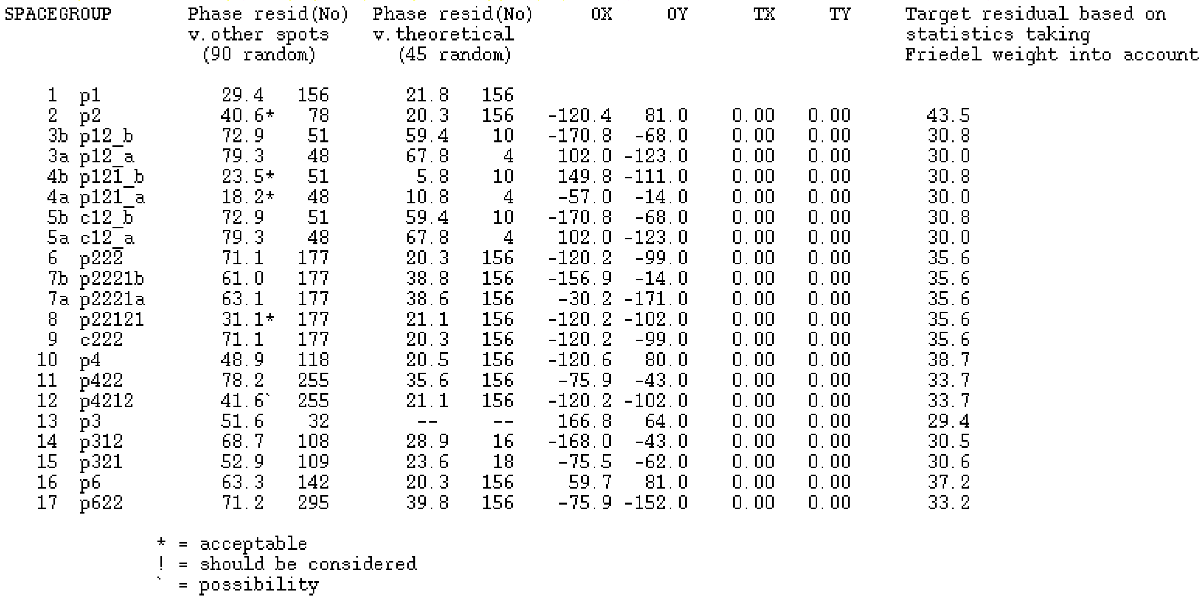


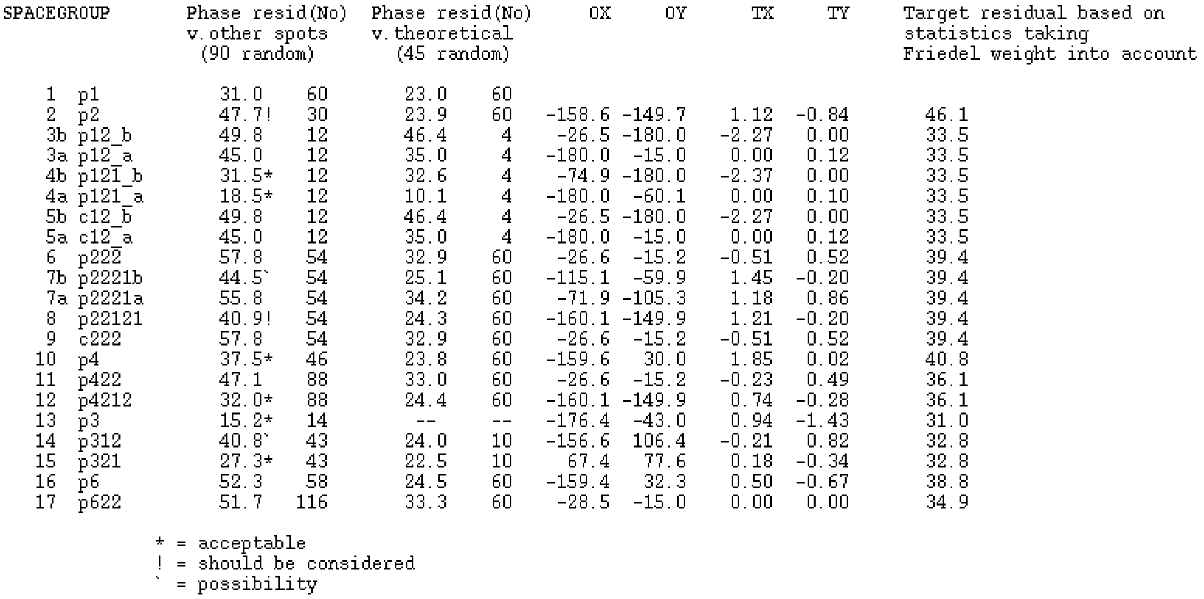
Figure 2.19: AFM Detail

The image shows the border of a crystalline area. On the left side, the crystal is still intact, whereas on the right side, single monomers start do disassociate from the crystal.



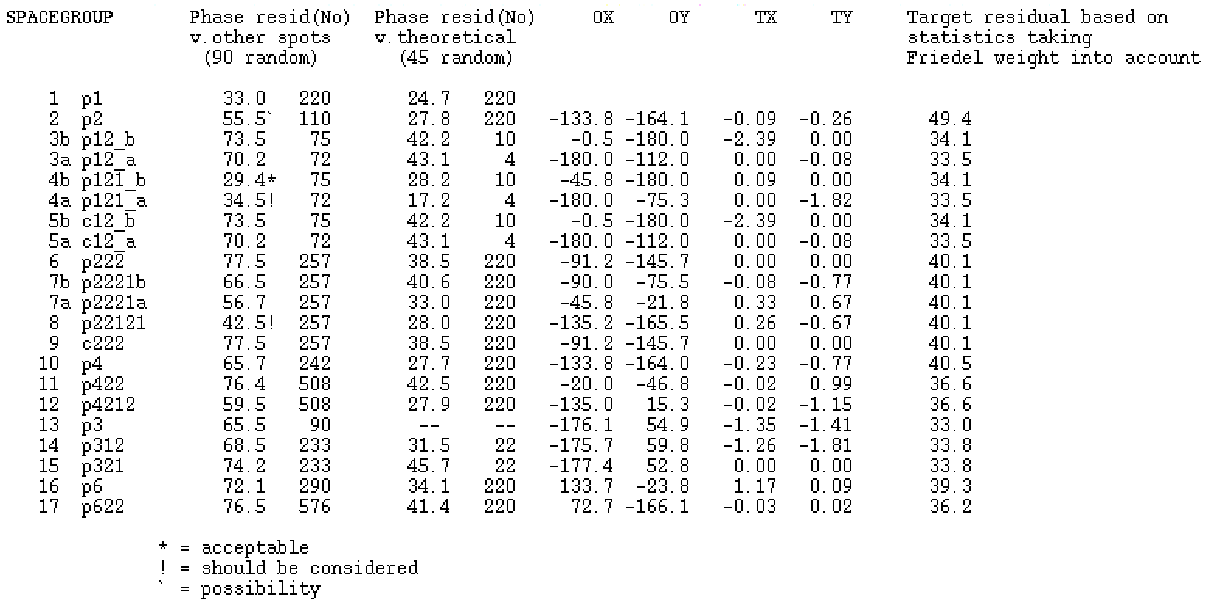
```
SPACEGROUP        Phase resid(No)  Phase resid(No)     OX      OY       TX      TY      Target residual based on
                  v.other spots    v.theoretical                                        statistics taking
                   (90 random)       (45 random)                                        Friedel weight into account

    1  p1            29.4   156      21.8   156
    2  p2            40.6*   78      20.3   156   -120.4    81.0     0.00    0.00        43.5
   3b p12_b          72.9    51      59.4    10   -170.8   -68.0     0.00    0.00        30.8
   3a p12_a          79.3    48      67.8     4    102.0 -123.0     0.00    0.00        30.0
   4b p121_b         23.5*   51       5.8    10    149.8 -111.0     0.00    0.00        30.8
   4a p121_a         18.2*   48      10.8     4    -57.0   -14.0     0.00    0.00        30.0
   5b c12_b          72.9    51      59.4    10   -170.8   -68.0     0.00    0.00        30.8
   5a c12_a          79.3    48      67.8     4    102.0 -123.0     0.00    0.00        30.0
    6  p222          71.1   177      20.3   156   -120.2   -99.0     0.00    0.00        35.6
   7b p2221b         61.0   177      38.8   156   -156.9   -14.0     0.00    0.00        35.6
   7a p2221a         63.1   177      38.6   156    -30.2 -171.0     0.00    0.00        35.6
    8  p22121        31.1*  177      21.1   156   -120.2 -102.0     0.00    0.00        35.6
    9  c222          71.1   177      20.3   156   -120.2   -99.0     0.00    0.00        35.6
   10  p4            48.9   118      20.5   156   -120.6    80.0     0.00    0.00        38.7
   11  p422          78.2   255      35.6   156    -75.9   -43.0     0.00    0.00        33.7
   12  p4212         41.6`  255      21.1   156   -120.2 -102.0     0.00    0.00        33.7
   13  p3            51.6    32       --     --    166.8    64.0     0.00    0.00        29.4
   14  p312          68.7   108      28.9    16   -168.0   -43.0     0.00    0.00        30.5
   15  p321          52.9   109      23.6    18    -75.5   -62.0     0.00    0.00        30.6
   16  p6            63.3   142      20.3   156     59.7    81.0     0.00    0.00        37.2
   17  p622          71.2   295      39.8   156    -75.9 -152.0     0.00    0.00        33.2

             * = acceptable
             ! = should be considered
             ` = possibility
```

Figure 2.20: ALLSPACE Statistics of one KdgM Image



```
SPACEGROUP        Phase resid(No)  Phase resid(No)     OX      OY       TX      TY      Target residual based on
                  v.other spots    v.theoretical                                        statistics taking
                   (90 random)       (45 random)                                        Friedel weight into account

    1  p1            31.0    60      23.0    60
    2  p2            47.7!   30      23.9    60   -158.6 -149.7     1.12   -0.84        46.1
   3b p12_b          49.8    12      46.4     4    -26.5 -180.0    -2.27    0.00        33.5
   3a p12_a          45.0    12      35.0     4   -180.0   -15.0     0.00    0.12        33.5
   4b p121_b         31.5*   12      32.6     4    -74.9 -180.0    -2.37    0.00        33.5
   4a p121_a         18.5*   12      10.1     4   -180.0   -60.1     0.00    0.10        33.5
   5b c12_b          49.8    12      46.4     4    -26.5 -180.0    -2.27    0.00        33.5
   5a c12_a          45.0    12      35.0     4   -180.0   -15.0     0.00    0.12        33.5
    6  p222          57.8    54      32.9    60    -26.6   -15.2    -0.51    0.52        39.4
   7b p2221b         44.5`   54      25.1    60   -115.1   -59.9     1.45   -0.20        39.4
   7a p2221a         55.8    54      34.2    60    -71.9 -105.3     1.18    0.86        39.4
    8  p22121        40.9!   54      24.3    60   -160.1 -149.9     1.21   -0.20        39.4
    9  c222          57.8    54      32.9    60    -26.6   -15.2    -0.51    0.52        39.4
   10  p4            37.5*   46      23.8    60   -159.6    30.0     1.85    0.02        40.8
   11  p422          47.1    88      33.0    60    -26.6   -15.2    -0.23    0.49        36.1
   12  p4212         32.0*   88      24.4    60   -160.1 -149.9     0.74   -0.28        36.1
   13  p3            15.2*   14       --     --   -176.4   -43.0     0.94   -1.43        31.0
   14  p312          40.8`   43      24.0    10   -156.6   106.4    -0.21    0.82        32.8
   15  p321          27.3*   43      22.5    10     67.4    77.6     0.18   -0.34        32.8
   16  p6            52.3    58      24.5    60   -159.4    32.3     0.50   -0.67        38.8
   17  p622          51.7   116      33.3    60    -28.5   -15.0     0.00    0.00        34.9

             * = acceptable
             ! = should be considered
             ` = possibility
```

Figure 2.21: ALLSPACE Statistics of one KdgN Image



```
SPACEGROUP        Phase resid(No)  Phase resid(No)     OX      OY      TX     TY      Target residual based on
                  v.other spots    v.theoretical                                      statistics taking
                   (90 random)       (45 random)                                      Friedel weight into account

   1  p1             33.0   220       24.7   220
   2  p2             55.5`  110       27.8   220   -133.8 -164.1   -0.09  -0.26        49.4
   3b p12_b          73.5    75       42.2    10     -0.5 -180.0   -2.39   0.00        34.1
   3a p12_a          70.2    72       43.1     4   -180.0 -112.0    0.00  -0.08        33.5
   4b p121_b         29.4*   75       28.2    10    -45.8 -180.0    0.09   0.00        34.1
   4a p121_a         34.5!   72       17.2     4   -180.0  -75.3    0.00  -1.82        33.5
   5b c12_b          73.5    75       42.2    10     -0.5 -180.0   -2.39   0.00        34.1
   5a c12_a          70.2    72       43.1     4   -180.0 -112.0    0.00  -0.08        33.5
   6  p222           77.5   257       38.5   220    -91.2 -145.7    0.00   0.00        40.1
   7b p2221b         66.5   257       40.6   220    -90.0  -75.5   -0.08  -0.77        40.1
   7a p2221a         56.7   257       33.0   220    -45.8  -21.8    0.33   0.67        40.1
   8  p22121         42.5!  257       28.0   220   -135.2 -165.5    0.26  -0.67        40.1
   9  c222           77.5   257       38.5   220    -91.2 -145.7    0.00   0.00        40.1
  10  p4             65.7   242       27.7   220   -133.8 -164.0   -0.23  -0.77        40.5
  11  p422           76.4   508       42.5   220    -20.0  -46.8   -0.02   0.99        36.6
  12  p4212          59.5   508       27.9   220   -135.0   15.3   -0.02  -1.15        36.6
  13  p3             65.5    90        --     --   -176.1   54.9   -1.35  -1.41        33.0
  14  p312           68.5   233       31.5    22   -175.7   59.8   -1.26  -1.81        33.8
  15  p321           74.2   233       45.7    22   -177.4   52.8    0.00   0.00        33.8
  16  p6             72.1   290       34.1   220    133.7  -23.8    1.17   0.09        39.3
  17  p622           76.5   576       41.4   220     72.7 -166.1   -0.03   0.02        36.2

             * = acceptable
             ! = should be considered
             ` = possibility
```

Figure 2.22: ALLSPACE Statistics of one NanC Image

NanC As it was the case for KdgN, the symmetry determination of NanC was less clear than that of KdgM. $p22_12_1$ and its sub symmetries had the best values for all the images. The high values for the phase error even for these symmetries (see figure 2.22) can be explained by the probably non perfect arrangement of the monomers in the crystals. A similar effect was also observed for the HasAR crystals (see section 3.6 for more details).

2.3.7.3 Determination of the β-barrel Periodicity

Motivation The projection maps of the three porins showed individual density blobs arranged in an elliptical shape. To check weather there is a periodicity in these blobs, out of which the number of individual β-strands could be deduced a periodicity analysis was performed.

Materials and Methods In order to identify the number of individual density blobs in the outer ring, and hence to possibly detect the number of β-strands forming the barrel, the following orientational correlation algorithm was applied: One of the symmetry related porins was cut out and the large and small axis of the ellipse were determined manually. The axis difference was corrected and the image was transformed to polar coordinates. The image was radially averaged and then Fourier transformed to estimate the periodicity. These steps were performed using the IPLT image processing toolbox [37].

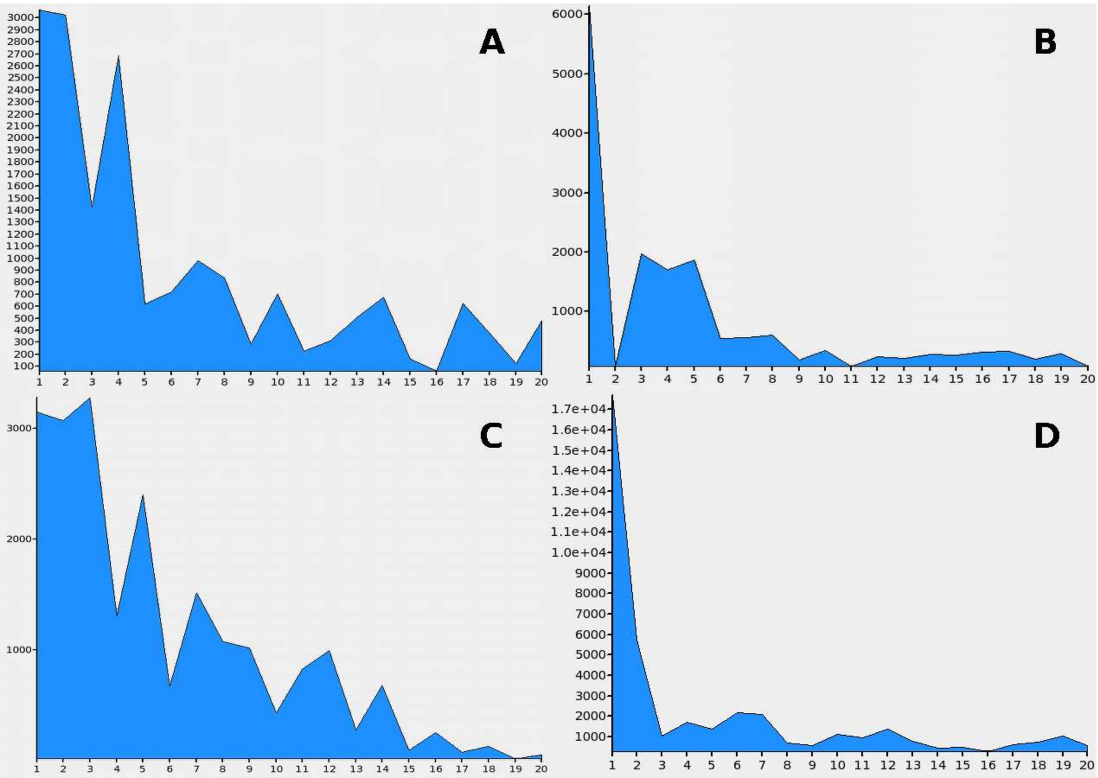


Figure 2.23: Periodicity Analysis

A) KdgM, B) NanC, C) KdgN and D) OmpF. The plots show the Fourier spectrum of the radial averages of the ring part of the projected monomers. A high value indicates that the monomer shows the respective rotational symmetry.

Results and Conclusions The Fourier spectrum of the radial averages of the ring part of KdgM, KdgN, NanC and OmpF (see figure 2.23) could potentially help us in the interpretation of the densities determined in the projection maps. The periodicity analysis of KdgM showed significant signals for 4, 7, 10, 14 and 17 fold periodicity, the analysis of KdgN for 5, 7, 12 and 14 fold periodicity, NanC for 3, 4 and 5 and the periodicity analysis of the calculated OmpF projection showed signals for 4, 6, 7, 10 and 12 fold periodicity, but not for the 16 fold periodicity. This results show that, a direct correspondence between density and β-strand is questionable. This can be explained by the observation, that β-strands are usually tilted (in average about 45° [44]) in the membrane and do not lead to clearly distinguishable density blobs in a 0° projection as can be seen in the calculated projections of OmpG and OmpF (see figures 4D and 4E in the porin paper presented in section 2.3.6). This contrasts with α-helices that not only insert more perpendicular to the membrane plane but also exhibit a more compact density. The number of β-strands in the three porins as determined with the periodicity analysis does not match the secondary structure prediction performed. The claimed prediction accuracy of the algorithms from [3] and [32] nevertheless favors the HMM models of 12 trans-membrane β-strands in all three proteins.

2.3.8 Conclusions and Outlook

The screening of the crystallization conditions using the cyclodextrin and the dialysis method yielded crystals suitable for image processing and the projection maps published in the paper described in section 2.3.6 give a first insight to the structure of these proteins. In order to get better crystals suitable for 3D reconstruction, the further screening of conditions needs to be very systematic. The proteins have proved to be very sensitive to small changes in conditions but KdgN for instance has also shown the capability to crystallize at very distinct conditions.

As the first crystallization trials were also part of the cyclodextrin method development, not all the experiments were conducted in a totally systematic way. In a future screening, be it to find better conditions for these presented proteins or for any other protein with unknown crystallization conditions, the experiments should be planed and performed in a much more systematic way: In a first series of experiments, lipids allowing the insertion of the protein need to be found. Then the correct LPR and a suitable detergent need to be determined. From this, one should start experimenting with different salts, salt concentrations and pH's to allow crystal contacts between the single protein molecules to form. All these steps need to be repeated several times, allowing a refinement of the conditions on one side, but also to test completely new combinations on the other side. The screening of the conditions at the microscope needs to be done with great care, specially in the first experiments. Once crystals have been detected, the conditions need to be refined, to produce the best crystals possible.

Chapter 3

Image and Data Processing

3.1 What We Had

3.1.1 MRC on SGI

Image and data processing of 2D crystals in our group based on the MRC image processing programs described in [9]. MRC evolved from a set of programs written 1970 [12] for the analysis of helical particles, icosahedral viruses [8] and later two-dimensional crystals [21]. Since this system evolved in an organic manner and many people contributed in an unsupported manner to it, MRC has evolved to a straggling mess of overlapping functions in some of the programs. Nevertheless, the system is the *quasi-standard* for the digital image processing of data collected in electron microscopy.

The system of standalone programs was extended and complemented by various people of our group (mainly Henning Stahlberg and Andreas Schenk) to suit our needs: A unified meta-data representation (i.e. the DATAFILE.dat file) was introduced, the single processing steps were combined in scripts to automate the processing to a certain extent, scripts for merging data from a defined directory structure were generated and the interaction with other programs such as CCP4 [1] was simplified.

Our running system was adapted to the SGI machines available and part of the programs used for the processing, such as the XIMDISP, would not run smoothly on machines with a different operating system.

3.1.2 IPLT

IPLT (**i**mage **p**rocessing **l**ibrary and **t**oolbox) is an open-source project for the image processing of data collected with electron microscopes, with a particular focus on electron crystallography(see [38] and [37]). The project was set up by Ansgar Philippsen and is under constant development. Great effort has been taken to make the system collaborative and expandable. The program code can be stored in and downloaded from a central repository, so that every developer can work with the current state of the system.

Since the current development focuses on the processing of diffraction data, IPLT could not replace the whole extent of MRC programs. There was a need to make an interface between the powerful (but often chaotic) programs of MRC and the graphical capabilities of the GUI from IPLT to continue and improve the image processing also on Linux systems.

3.2 Improvements in the Image Processing

3.2.1 MRC on Linux

Computers running on a Linux operating system are becoming more and more ubiquitous and are by now the quasi standard for scientific computing. For this reason and also in terms of available processing power in our lab, we wanted to have our MRC system also available in Linux. The official Linux port available from MRC was the base of our new system, and we had to adapt and port our various additions and modifications made over the last years. Most of the port was straight forward, but at some places problems emerged, since not all MRC programs were running properly on the new operating system. XIMDISP for example could not be used our Linux machines and due to its inconvenience in the user interaction, it was skipped in favor of new developed GUI in IPLT (see below).

3.2.1.1 The New Meta-Data Format: imagedata.xml

The handling of meta-data in our MRC setup was rendered compatible with the IPLT XML-info implementation [37]. The file *imagedata.xml* replaces the *DATAFILE.dat*. As with the old format, data used in the different MRC programs can be read from this file and data resulting from calculations can be written in this file. The big advantage of the new XML-file is the hierarchical organization of the data, i.e. one can now store information valid for the whole project or for a certain tilt angle in a *parent* xml-file which can be accessed through the files containing the information specific for each image.

A python script (*dat2xml.py*) was generated to convert data from the old format into the new format, allowing the user to seamlessly continue the work from older projects on the new system.

3.2.2 Interaction Between MRC and IPLT

A whole set of programs for image processing of electron micrographs is unified in MRC. These programs were created over a time range of more than 30 years. In these programs there are a lot of great algorithms but also outdated code using heuristics that probably prevent to get access to the structural details in the high resolution range.

To establish a new and clean image and data processing scheme there is a need to evaluate all the algorithms and to decide whether they should be incorporated in the new processing, improved to fit the new capabilities computers have nowadays or skipped and replaced by other algorithms. Since it is not sensible to do this all at once, there has to be the possibility to do this step by step. First steps in this direction have been taken by porting MRC to Linux and by allowing its interaction with IPLT. IPLT is able to read and write the various file formats from MRC, so that the processing can be switched without bigger problems between IPLT and MRC. Once an algorithm is implemented in IPLT, data can be processed and compared to results of data processed with the corresponding MRC program.

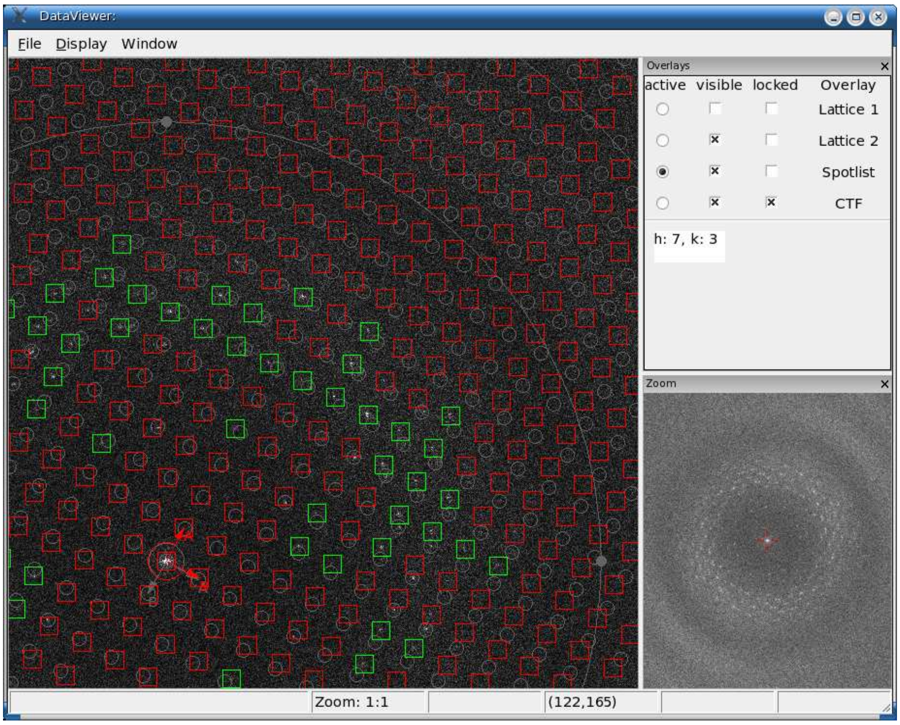
As a first step, the graphical user interaction of MRC (XIMDISP) was replaced with the overlays in IPLT (see below). Further steps include the replacement of the final calculation of the projection maps and the conversion of different file formats (i2i from IPLT is used to do this now). In a further development we can now solve the problems arising in the processing by MRC using IPLT routines and therefore replace MRC step by step, without loosing the overall power of it.

3.2.2.1 Contributions

IPLT as a collaborative system allows several ways of contributions. The first implementations in the GUI were done by Johan Hebert, a summer student from Stockholm. The actual implementation as an overlay framework was mainly done by Andreas Schenk and Ansgar Philippsen. My contribution was the implementation of the spot-list overlay with its internal representation of the spot-list, the latDet.py script and extensive testing and debugging of the GUI.

3.2.2.2 Overlays in the IPLT GUI

IPLT has evolved considerably over the last few years. One feature that proved to be of special interest was the addition of the overlay concept to the viewers in the GUI. An overlay is a graphical component that is added to the viewer displaying an image of any kind. One can interact with this overlay to get information from the underlying image or on the displayed overlay. The overlays scale and move with the image and behave like they are a part of the image itself. Every overlay can be chosen to be *active* and then be manipulated individually using the mouse and keyboard. The *inactive* overlays are still visible, but appear in a less prominent way (a weak grey) (See figure 3.1 for an example). The *lock* mechanism can be used to avoid change in a particular overlay. There is also a choice whether an overlay should be visible or not. These features allow the interaction with various overlays at one time, without overloading the viewer.



active	visible	locked	Overlay
○	☐	☐	Lattice 1
○	☒	☐	Lattice 2
●	☒	☐	Spotlist
○	☒	☒	CTF

h: 7, k: 3

Figure 3.1: Overlays in IPLT
Several overlays are visible in the main window. The overlays can be activated, set visible or locked from a control panel on the right of the viewer. The overlay properties (such as symbol size and shape) can be set by right-clicking on the respective overlay name.

3.2.2.3 Integration in the Work-flow: The latDet.py - Script

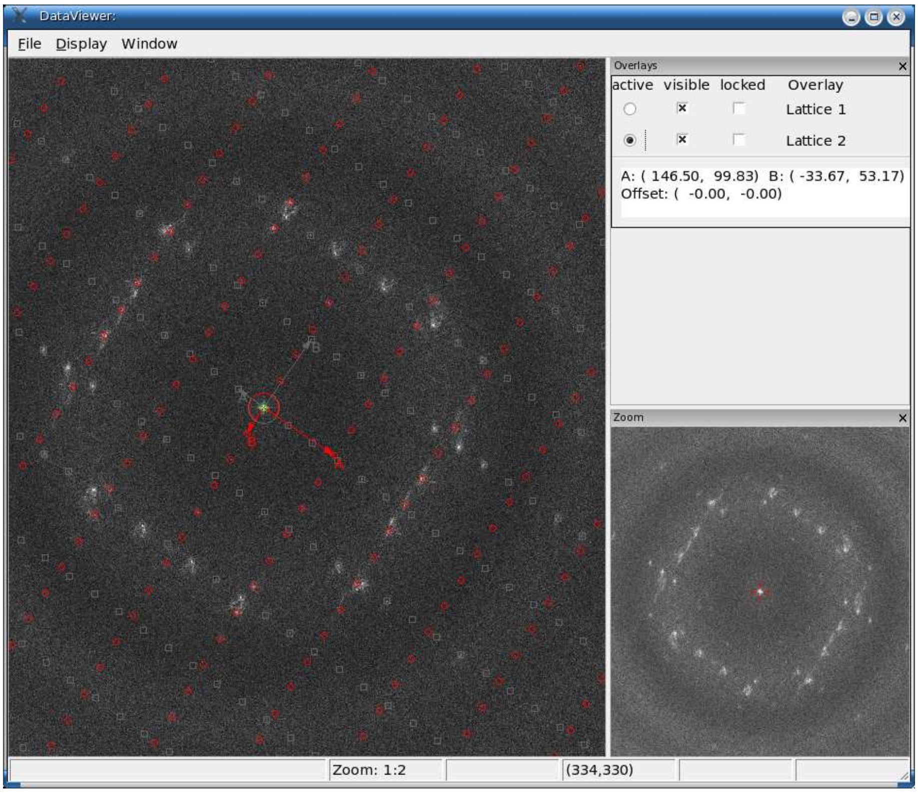
Our MRC setup is in principle able to perform a full automatic processing of a series of images once the basic parameters like unit cell vectors, magnification of the images, their sampling rate or the kind of image (negative stain or cryo electron microscopy) are known. Scripts were generated to handle the data flow between the various programs in MRC and to handle different images and the meta-data in the image directories. In practice however there are limitations in this automatic processing scheme, leading to the need of an intervention of a user through a graphical user interface (GUI).

Determination of the Lattice The determination of the lattice should work automatically, once the unit cell vectors and the tilt angle are known. In practice however, there were several problems preventing the automatic lattice determination from working correctly. One reason for this was that some of the peaks were weak or not visible at all due to cancellation resulting from the crystal symmetry. Further it was difficult to assign the lattice vectors in tilted images, since there, the distortion of the lattice vectors depends on the angle of the tilt taxis relative to the lattice. Another problem arose, when more than one crystal layer was present on one image. In that case the automatic detection can not tell apart spots from different lattices. The situation of having more than one lattice arose when the crystal consisted of collapsed tubes or vesicles (as it is the case in KdgM, NanC and KdgN see section 3.5 for a detailed discussion) or when more than one crystal was present in the selected area. In the later case, the easiest way would be to cut the image and to process the images individually.

A way to circumvent this problem of the automatic assignment of multiple lattices was to determine the lattices by hand using one *lattice overlay* per lattice for its indexing. An example of an image with two determined lattices can be seen on figure 3.2. The image shows the calculated Fourier transform of an image of a KdgM tube. The possibility to determine two or more lattices in parallel made it easy to keep an overview on which spots belong to which lattice and whether there were still spots that do not fit into any of the lattices.

Manual Defocus Determination The defocus of an image can be determined automatically but this procedure is very time consuming and in some cases, the calculation can lead to a completely wrong result. To correct for wrongly determined defocus values or to estimate initial defocus values, without having to run the calculation, one can fit the concentric zero lines of the CTF into the Thon rings of the image using *a CTF overlay* (see figure 3.3).

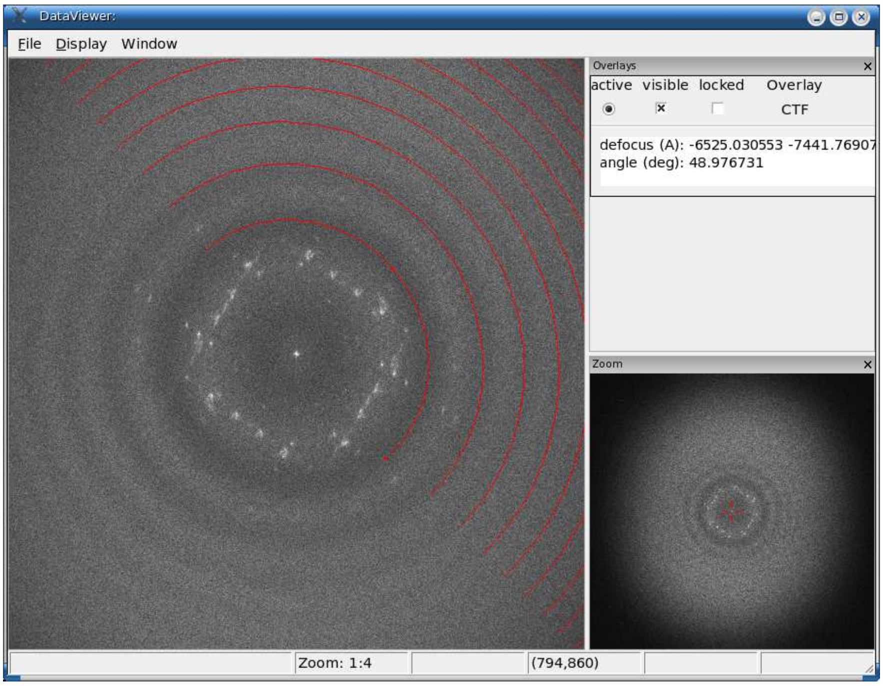
Selection of an Initial Spot-list MRC needs an initial spot-list to generate a reference for the subsequent processing steps. This selection could be done automatically by selecting the closest indices's from a given lattice but, as the quality of the following steps depends to a great extent on the choice of this initial reference, a manual selection of good spots for the initial spot-list is sensible. The *spot-list*



active	visible	locked	Overlay
○	☒	☐	Lattice 1
●	☒	☐	Lattice 2

A: (146.50, 99.83) B: (-33.67, 53.17)
Offset: (-0.00, -0.00)

Figure 3.2: Two Indexed Lattices
On this calculated Fourier transform of a KdgM tube, two lattices are determined using two lattice overlays. The user can switch between the two lattice to improve the indexing.

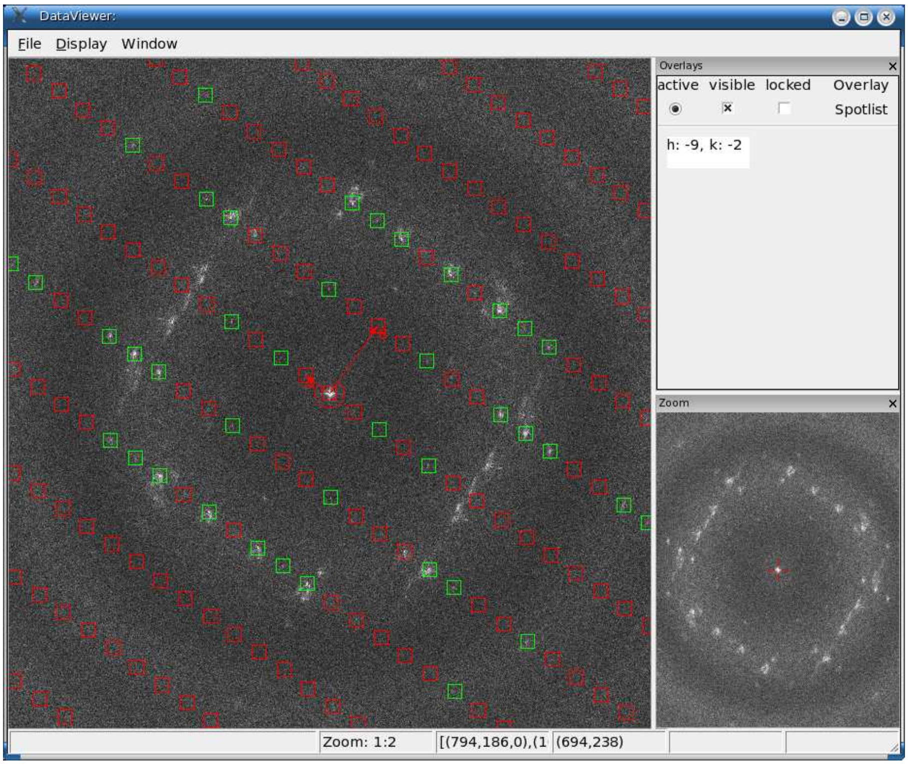


active	visible	locked	Overlay
●	☒	☐	CTF

defocus (A): -6525.030553 -7441.76907
angle (deg): 48.976731

Figure 3.3: CTF Overlay.

The CTF is fitted to the zero lines in the power spectrum to determine the defocus of an image. The aspect ratio can be changed if there is astigmatism, or kept constant if the defocus is the same in both directions. Depending on the users selection the CTF overlay can display the full rings, or as shown, only the half rings.



active	visible	locked	Overlay
●	☒	☐	Spotlist

h: -9, k: -2

Zoom: 1:2 | [(794,186,0),(1 | (694,238)

Figure 3.4: Spot-list Overlay.
The selected spots are displayed in green, all the other lattice points in red. A spotlist can be saved to or loaded from a file.

overlay was created to select the best spots manually from a projected lattice. That way one can ensure that only the good spots are used for the generation of the initial reference.

This spot-list was implemented as an overlay taking a lattice as input parameter and returning a list of h-k indices's. The spots in the list are highlighted and can be added to or removed from the list by Ctrl-clicking on them. The not-selected lattice indices's are displayed without highlighting. The spot-list can be saved to or loaded from a file or it can be emptied completely to restart the selection. In figure 3.4 there is an example of an active spot-list, where already some spots have been selected.

The latDet.py Script The steps described above have to be done for every image one wants to process. To facilitate the user interaction and to save typing effort I generated a Python script, that provides all the required functions. The functions are mirrored into a click-able interface, which only allows to perform the sensible steps at a given moment. For example you can not save a lattice before you have determined it (see figure 3.5A).

The interface of the latDet.py script can be seen in figure 3.5. (A) shows the initial state of the interface, (B) the state of the interface, after the first lattice is displayed and the spot-list has been created. When the script is initiated, it reads in the basic values from the imagedata.xml (image name and sampling rate) and the Fourier transform of the image is loaded and displayed.

The elements in the interface call functions in the Python script:

- *Display first lattice* and *Display second lattice*: Reads the first or the second lattice from the imagedata.xml, generates a LatticeOverlay and adds it to the viewer displaying the Fourier transform of the image. If no lattice is determined yet, a default lattice is displayed.
- *Reload first lattice* and *Reload second lattice*: Reads the lattice vectors from imagedata.xml and sets the values of the lattice overlay already on display.
- *Save first lattice at position 1, Save first lattice at position 2, Save second lattice at position 1* and *Save second lattice at position 2*: Reads the values from the displayed LatticeOverlay and writes them into the imagedata.xml. Position 1 is the lattice used for the processing, position 2 the lattice used for exclusion of spots that are to close together.
- *Create a spotlist*: Creates a spotlist overlay with the first lattice from the imagedata.xml and an empty spotlist.
- *Load a spotlist*: Creates a spotlist overlay with the first lattice from the imagedata.xml and the spotlist saved in the .spt file.
- *Save a spotlist*: Saves the determined spots (as h-k-indices's) into the .spt file. The lattice coming from the imagedata.xml can not be changed with this overlay and is thus not saved.
- *Create a CTF*: Creates a CTF overlay with default parameters and adds it to the viewer.
- *Load a CTF*: Reads the defocus parameters from the imagedata.xml and generates and displays a CTF overlay.
- *Save a CTF*: Writes the determined defocus parameters to imagedata.xml.

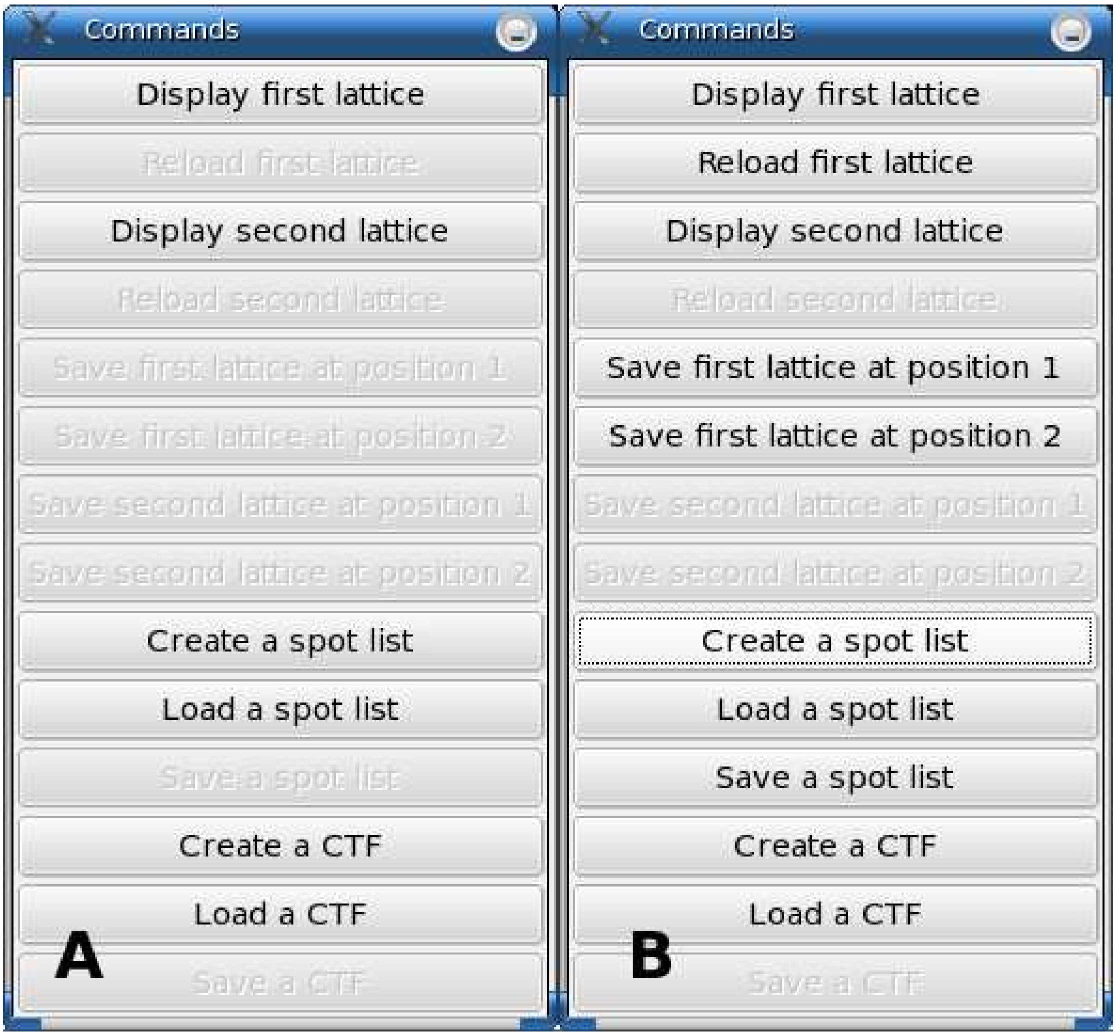


Figure 3.5: latDet Interface.
The interface allows only the sensible actions. A) Initial state of the interface after starting the script and B) interface after the first lattice has been displayed and the spotlist has been created.

3.3 IPLT Publication 07 (1)

IPLT is in ongoing development. A publication from the year 2007 ([37]) resumed the latest development. It was published in the *Journal of Structural Biology* under the title ’Collaborative EM image processing with the IPLT image processing library and toolbox’. It shows what developments and changes in architecture have been achieved since the last publication on IPLT three years ago [38].

3.3.1 Contributions

Ansgar Philippsen is the driving force behind IPLT. He is first author and responsible for the general planing and for a lot of implementations. Andreas Schenk was the other main contributor. My contribution to this paper is based on the implementations of algorithms. I also was part of the ongoing discussions on the general logic of IPLT and on specific problems arising during the implementation of algorithms or procession paths.

3.3.2 The Paper

Following is the publication "Collaborative EM image processing with the IPLT image processing library and toolbox" as published in the Journal of Structural Biology.

Available online at www.sciencedirect.com
ScienceDirect
Journal of Structural Biology 157 (2007) 28–37
Journal of Structural Biology
www.elsevier.com/locate/yjsbi

Collaborative EM image processing with the IPLT image processing library and toolbox ☆

Ansgar Philippsen *, Andreas D. Schenk, Gian A. Signorell, Valerio Mariani, Simon Berneche, Andreas Engel

Maurice E. Müller Institute for Structural Biology, Biozentrum Basel, Switzerland

Received 12 April 2006; received in revised form 13 June 2006; accepted 15 June 2006
Available online 14 July 2006

Abstract

We present the Image Processing Library and Toolbox, IPLT, in the context of a collaborative electron microscopy processing effort, which has driven the evolution of our software architecture over the last years. The high-level interface design as well as the underlying implementations are described to demonstrate the flexibility of the IPLT framework. It aims to support the wide range of skills and interests of methodologically oriented scientists who wish to implement their ideas and algorithms as processing code.

Keywords: Image processing; Electron crystallography; Software engineering

1. Introduction

Structural investigations of biological samples with the electron microscope is a continuously growing and evolving field. The analysis of the measurements to extract the structural information is referred to as image processing, because a collection of images forms the majority of raw data. The 3D EM community has produced, and is producing, a large collection of sophisticated image processing algorithms, made available in several software packages. Here we describe one of these software packages, the Image Processing Library and Toolbox (IPLT), that was started in our group, with a special emphasis on the dynamic nature of the 3D EM research.

In our first paper on IPLT (Philippsen et al., 2003), we discussed the rational for creating a new image processing package from scratch, our intended goals, and the architectural design. We wish to provide a modular, integrated, collaborative, flexible, extendable, open-source, cross-platform framework for image processing of electron microscope images. Modular in the sense that it is build from clearly separated building blocks that may be used together as seen fit; integrated implying that it is a single package whose components have been made to work together; collaborative meaning that we do not want it to become an in-house software only but evolving by means of community involvement; flexible because it accommodates the various user requirements; extendable hinting at the intended ease of adding algorithms and procedures.

This manuscript continues pretty much where the previous one left off at, namely how the concepts and the first implementation of the IPLT framework have proven themselves within a, albeit small, collaborative environment, in particular describing the maturation of the architecture and the currently implemented image processing modules. This is preceded by a recapitulation of the major features as well as a discussion of the implementation.

To allow a more concise description of the current state of IPLT, the software engineering terms user, abstraction, interface, encapsulation, implementation, instance, class, and object are introduced. Since these are all coupled, a suitable definition is the generic statement: An interface offers an abstract interaction of some encapsulated functionality to a user, who does not need to know any details

---

☆ The IPLT homepage is found at http://www.iplt.org.
* Corresponding author. Fax: +41 61 267 2109.
E-mail address: ansgar.philippsen@unibas.ch (A. Philippsen).

1047-8477/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jsb.2006.06.009

of the actual implementation of this functionality. In terms of software development, an abstract concept—like an image—can be implemented as a set of classes in the context of a specific design; each class has a set of methods, which form the class interface; each method has its (hidden) implementation in the programming language, which in turn is an abstraction to the CPU instruction set and implemented in a compiler or interpreter. On the opposite end, an overall image processing scheme is implemented in terms of conceptual steps, each in turn leading to concrete use of the various components of a class library. And finally, the processing application may be completed by adding a graphical user interface.

In an object oriented language, the construct that offers this encapsulation scheme is usually referred to as *class*, and each instance, i.e. independent unit, of a class is usually referred to as *object*. The distinction between class and object may not seem obvious now, and there are certainly many contexts where the terms can be interchanged. Here, we use class when we are talking about a design construct, and object as the actual use of the class.

While the concepts of interface and implementation have been around implicitly since the dawn of programming, it is only with the advent of object oriented design that they have been explicitly formulated, in particular discussing the difficulties and challenges of designing a good interface, which is an art in itself. These concepts have proven to be especially powerful in an environment of larger projects, with evolving software and heterogeneous developers, exemplified by techniques such as refactoring/extension cycles (Fowler, 1999) or Extreme Programming (Beck and Andres, 2004).

In the following sections, we describe the interfaces available in IPLT, descend into some implementation details where appropriate, and explain the rational for this particular design. While this manuscript, like the last one, is still aimed at the interested developer, and not so much the regular user, one of the major claims contained herein is that the particular design of the IPLT empowers a casual user to become a contributor, due to the various framework layers that range from simple Python scripts to sophisticated C++ routines.

2. Technical description

Since the publication of the first paper on IPLT (Philippsen et al., 2003), the software has evolved considerably, not only in the actual processing capabilities, but also in the software engineering sense, mainly due to its direct involvement in a collaborative effort to establish a novel 2D electron crystallography processing suite (see below). As a consequence, several previously described concepts have been superseded, some design decisions were discarded and replaced with more sophisticated ones. These changes affect both the interface as well as the implementation level. Future changes, however, anticipate the integration of new ideas without affecting the interfaces to ensure a stable evolution of the system.

The current software architecture of IPLT is schematized in Fig. 1. It comprises C++ and Python components, where the basic constituents are written in C++ and consequently reflected into Python by means of a wrapper. Conceptually, the C++ level is meant to provide the basic building blocks, and the processing logic is then implemented on the Python level, utilizing these building blocks.

2.1. Images

The **`ImageHandle`** class represents the high-level, central image concept in IPLT. It combines several features and functionalities: First of all, it offers a unified interface to the underlying image state (explained below) and its properties, independent of the actual state used, thereby exempting the users from dealing with all possible state combinations and their instances. This unified interface allows access to the image values as well as retrieval of the essential image properties extent, pixel sampling, type, and domain. Second, the **`ImageHandle`** interface provides a set of **`Apply`** methods which facilitate the interaction with all algorithm objects; in a nutshell, each algorithm can be applied either in-place or out-of-place, independent of the actual algorithm object implementation. Third, in its nature as a handle, it removes all memory management responsibilities from the user: a handle only points to an automatically allocated image state instance, and copying handles will not duplicate this memory area, but it will be shared by all handles, and de-allocated when no longer referred to. This feature has the additional advantage that code in C++ and Python is almost identical, save for syntax differences (see Fig. 2). Fourth, it supports the unary or binary operators negation, addition, subtraction, multiplication, and division, which operate on a per-pixel level.

In view of all this comfort and sophistication, there is but a single drawback of the image handle, namely that reading and writing of pixel values is relatively slow compared to directly accessing the memory. As a consequence, a special mechanism has been implemented to give developers access to the particular image state of an image handle, by means of an image state algorithm—as explained further below.

The image state represents the combination of a particular value storage or pixel type, namely real or complex, with a particular domain, namely spatial, frequency, or half-frequency. It manages the actual pixel value memory, and allows direct, raw access to it. Readers familiar with such a scheme will immediately notice that the template feature of C++ is perfectly suited for implementing such a state ensemble, and this is indeed how it is done. A full description of the templated implementation is, however, beyond the scope of this paper. It must suffice to mention that the particularities of our architecture (mostly the hybrid implementation of C++ and Python as well as the strong modularity) render the use of templates complicated. For this reason, the complexity of dealing with templates has

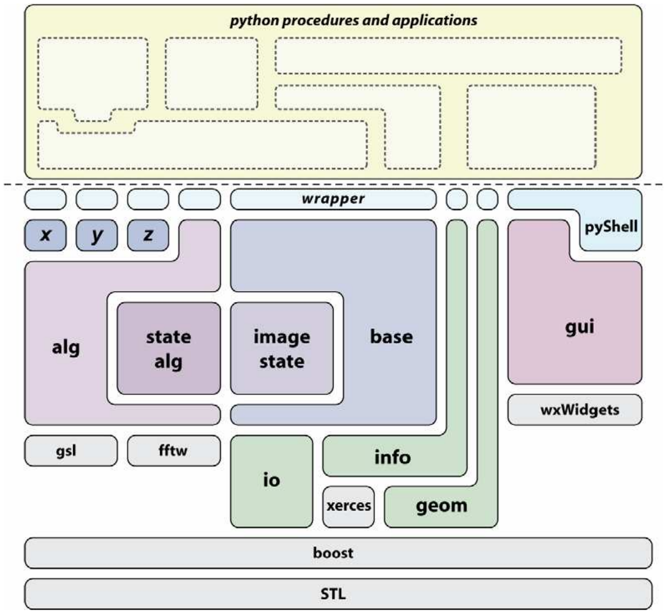


Fig. 1. Current IPLT software architecture. This scheme depicts the current components that comprise the IPLT architecture. The main components are implemented in C++, and are made available in Python by means of a wrapper. At the heart of IPLT lies the base class library, with the embedded image state module. The image algorithm class library also contains an embedded module for the image state algorithm base classes. Concrete algorithms, denoted by *x*,*y*, and *z*, build upon these base classes. The graphical user interface is a separate component, tightly coupled to the python layer by the python shell widget, as explained in the text. Helper libraries are either part of IPLT (io, info, and geom—indicated in green), or are open-source projects (indicated in grey). The processing procedures are to be mostly implemented in Python, as indicated by the stippled yellow blocks.

```
                 Python                |          C++
```

Load an image from a file and store it in an image handle:

```
im1 = LoadImage("file")                | ImageHandle im1 = LoadImage("file");
```

Create an instance of the statistics algorithm, apply the statistics algorithm to the image, and etrieve a result, in this case the mean, from the algorithm object:

```
stat = alg.Stat()                      | alg::Stat stat = alg::Stat();
im1.Apply(stat)                        | im1.Apply(stat);
mean = stat.GetMean()                  | double mean = stat.GetMean();
```

Extract the extent from the first image and use it to create a second image of the same size:

```
ext1 = im1.GetExtent()                 | Extent ext1 = im1.GetExtent();
im2 = CreateImage(ext1)                | ImageHandle im2 = CreateImage(ext1);
```

Randomize the values in the second image by applying an instance of the randomization algorithm object, which in this case is created on the fly, since it is not needed afterwards:

```
im2.ApplyIP( alg.Randomize() )         | im2.ApplyIP( alg::Randomize() );
```

Add the two images together, as easily as adding numbers:

```
im3 = im1 + im2;                       | ImageHandle im3 = im1 + im2;
```

Manually copy all values from one image to the other (this only serves to illustrate the syntax, there are faster ways to do so!):

```
for p in ExtentIterator(ext1):         | for(ExtentIterator p(ext1); !p.AtEnd(); ++p)
                                       | {
    val = im3.GetReal(p)               |     Real val = im3.GetReal(p);
    im1.SetReal(p, val)                |     im1.SetReal(p, val);
                                       | }
```

Fig. 2. General code example. This code example is meant to demonstrate how images and algorithm objects are used in practice, in both Python (left) and C++ (right). Apart from syntactical differences, the usage is the same in both languages.

been encapsulated as much as possible in the image state algorithms, so that algorithm developers will be only marginally confronted with them.

2.2. Functions

A concept related to the image is the function, which differs in one important feature: it does not contain any data storage, rather it calculates the requested values for a particular pixel position on the fly, and there is no method to set pixel values. Otherwise, the interface between image and function is identical. A developer building a concrete function derives from either the **`RealFunction`** or **`ComplexFunction`** base class, which ensure seamless integration with the framework.

2.3. Algorithm objects

The image interface is complemented by an assortment of image manipulation algorithms, which are implemented as independent algorithm objects. As the name suggests, each algorithm is implemented as a class, which is to be instanced as an object in order to apply the algorithm to an image. Both algorithm users as well as developers profit from this concept, as explained forthwith.

From the algorithm user's perspective, both in C++ and in Python, this offers some advantages over a more classical scheme, where an algorithm is simply a function: an object instance is stateful, meaning it can store parameters and/or results within its lifetime, abolishing the necessity to pass a long list of parameters upon the actual algorithm application. This results in a clearer and more intuitive interface of an algorithm compared to calling a subroutine or function, since the construction of an algorithm object can be reduced to the essential parameters, the rest being default ones. In addition, all parameters can be set by individual methods, and the result can be retrieved by clearly named methods as well. While this may take up more lines than a single function call, the single line constructs with cluttered parameters, half of them modified to pass the result back to the caller, are a thing of the past. Some fragments of code that illustrate this are given in Fig. 2.

From an algorithm developer's perspective, two main features need to be mentioned: First, each algorithm is a concrete class derived from one of three algorithm base classes, all of which integrate seamlessly with the image handle interface; in combination with the build system (see below), this allows each developer to concentrate only on the implementation of the actual algorithm, and the integration with IPLT on both the C++ as well as the Python level is taken care of by the framework. Second, it is via the special image stage algorithm classes that a developer gains fast access to the templated image state, while maintaining the various encapsulation layers of the framework.

The base classes have been modeled after three algorithm types that differ in the way they modify an image:

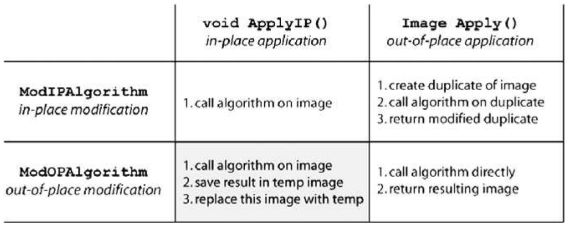
Non-modifying algorithm: Applies to both images and functions. Only needs to read values, but not change any. The result of the algorithm can be retrieved via its public interface. *Examples*: Statistics calculation, peak search.

In-place modifying algorithm: An algorithm that changes the image values without duplicating the image first. *Example*: Assigning a random value to each pixel.

Out-of-place modifying algorithm: An algorithm that is more easily implemented by creating a new image. *Examples*: Rotation, explicit convolution.

The usage of the algorithm proceeds via the image handle **`Apply`** methods (Fig. 2), and it must be mentioned that the in-place/out-of-place application is decoupled from its implementation, leading to a usage matrix shown in Fig. 3. As a consequence, the implementation may be done in the manner that the developer sees fit, leaving it up to the user to decide how the algorithm is applied in a particular situation.

The image state algorithms are always encapsulated by and accessed from a high-level algorithm, similar to the way that an image state is managed by an image handle. Technically, the image states are implemented as a templated class, while an image state algorithm contains a templated member function, whose template parameter match the ones from the image state class. As a consequence, the compiler will generate a member function in the algorithm object for each image state there is, and upon application of an algorithm to an image, the member function corresponding to the specific image state will be called, with that state as a parameter. In addition, template specialization can be used to write algorithms that apply only to certain type and domain combinations. Therefore, the developer can implement the algorithm once, in a generic, templated way, the compiler takes care of producing the code for each variant that is required, and the base class provides the seamless integration into the rest of the IPLT framework, as shown in Fig. 4. The same figure also demonstrates the straightforward way to wrap an existing function with an IPLT algorithm object; therefore, an algorithm does not need to be implemented from scratch if it is desirable to reuse a routine from an existing library.



	`void ApplyIP()` *in-place application*	**`Image Apply()`** *out-of-place application*
`ModIPAlgorithm` *in-place modification*	1. call algorithm on image	1. create duplicate of image 2. call algorithm on duplicate 3. return modified duplicate
`ModOPAlgorithm` *out-of-place modification*	1. call algorithm on image 2. save result in temp image 3. replace this image with temp	1. call algorithm directly 2. return resulting image

Fig. 3. Algorithm application matrix. In-place algorithms can be applied out-of-place and vice versa. Unavoidable memory duplication only takes place in the in-place application of an out-of-place algorithm (shaded in grey).

Imaginative procedural call to a frzl algorithm that requires a pointer to the image data, the width and the height of the image, as well as the two parameters a and b. If everything works ok, value 1 is returned as a status, the image is modified, and the result is stored in parameter c. As an additional detail, parameter b can be zero for most usage cases.

```
int frzl(double* i, int w, int h, double a, double b, double* c);
```

Wrapping of this function within IPLT

```
#include “frzl.h”

class FrzlFnc {
public:
  // initialization with a, b has a default of zero
  FrzlFunc(double a, double b=0.0): a_(a), b_(b), c_(0.0) {}

  // parameters and result interface
  void SetA(double a) { a_=a; }
  void SetB(double b) { b_=b; }
  double GetC() const { return c_; }

  // declaration of the templated function
  template<typename T, class D>
  void VisitState(iplt::ImageStateImpl<T,D>& state);

private:
  // stateful storage of parameters and results
  double a_, b_, c_;
}

// actual implementation of wrapping call in this case only for real/spatial images
template<>
void FrzlFnc::VisitState(iplt::ImageStateImpl<REAL,SPATIAL>& state)
{
  double c;
  int ret = frzl(state.GetDataPtr(),
                 state.GetExtent().GetWidth(),
                 state.GetExtent().GetHeight(),
                 a_, b_, &c);
  if(ret==1) {
    c_ = c;
  } else {
    throw Error(“Frzl call failed”);
  }
}

// this creates the actual algorithm object Frzl
typedef ImageStateModIPAlgorithm<FrzlFnc> Frzl;
```

Use of this function (in Python, would be almost identical in C++)

```
img_in = LoadImage("file")

# make frzl algorithm object, only specifying parameter a
frzl_alg = Frzl(2.0)

# apply and save in new image
img_out = img_in.Apply(frzl_alg)
c = frzl_alg.GetC()
```

Fig. 4. Algorithm wrapping example.

2.4. Graphical user interface

The GUI is a separate component of IPLT, meaning that it is non mandatory for normal usage. It consists mainly of two graphical components (called widgets): the Python shell and the image viewer. The Python shell offers a text widget with a python command line interpreter, into which both the IPLT modules and the python wrappers for the graphical user interface, such as the image viewer, have been preloaded.

The image viewer displays an image in a widget. Each viewer is an instance of a python object, since a viewer is opened via the command line in the graphical python shell. As a consequence, there are two interfaces: the graphical, mouse based one, and the methods of the viewer object instance. The latter makes it very easy to utilize input from the graphical user interface in processing scripts.

The functionality of the image viewer may be extended with a plugin mechanism, using a concept we have termed overlays. These overlays allow graphical elements, such as lines, to be drawn on top of the displayed image, as well as input events from the mouse and keyboard, targeted to the image viewer, to be intercepted. Overlays are derived from a base class in either C++ or Python, and an arbitrary number of them can be added to an image viewer, with one active overlay receiving the input events. Thus, the functionality of the image viewer window may be assembled dynamically on the Python level, as deemed necessary, with each overlay being independent of the other. (An example is given in Fig. 7.)

2.5. Info

Only a minimal set of properties is contained directly with the image (namely, as described above, extent, type, domain, and pixel sampling). All the remaining ones (such as sample number, magnification, tilt geometry, etc) are meta-data, represented by a tree-like *Info* hierarchy, as given in Fig. 5, where the nodes are *groups*, and the leafs are *items*. Groups and items are addressed by a path. Attributes may be assigned to both to specify additional information (such as physical units or versioning info). Items contain a value, with an interface for conversion to various types. This closely follows the setup of the EMDEP database (Tagari et al., 2002).

The info implementation is a thin wrapper to the XML-DOM representation (http://www.w3.org/XML), in this case provided by the Xerces library from the Apache project (http://www.apache.org). The high-level classes, **`InfoHandle`**, **`InfoGroup`**, and **`InfoItem`**, all utilize the handle concept, again (i) freeing the user from memory management tasks and (ii) leveling the C++/Python behavior.

In order to deal with project specific meta-data, a complete info hierarchy may be assigned as the default of another hierarchy. As a result, the request for a specific group or item will fallback to the default hierarchy if not found in the current one. A processing scheme can, for example, load a project specific xml file first, and then use it as a default for each image specific xml file.

2.6. Geom

A comprehensive C++ class library named **`geom`** offers matrices and vectors in 2, 3 or 4 dimensions, composite constructs like lines and planes, and relevant convenience routines, e.g. generation of transformation matrices, line-to-plane distance calculations, etc. It is also available on the Python level.

2.7. Import and export

Several common file formats can be imported and exported by the IO module, based on an abstract mechanism that allows straightforward addition of new IO plugins. The supported formats are MRC, CCP4, PNG, and TIFF (8 and 16 bit).

2.8. External libraries

The major components of IPLT rely on the C++ standard template library (Josuttis, 1998, http://www.sgi.com/tech/stl/), and on the following external (class) libraries: Python (http://www.python.org), boost (http://www.boost.org), Xerces-C (http://www.apache.org), wxWidgets (http://www.wxwidgets.org), FFTW (Frigo and Johnson, 1998, http://www.fftw.org) and the Gnu Scientific Library (http://www.gnu.org/software/gsl). All are open-source projects, popular, well supported, and cross-platform compatible.

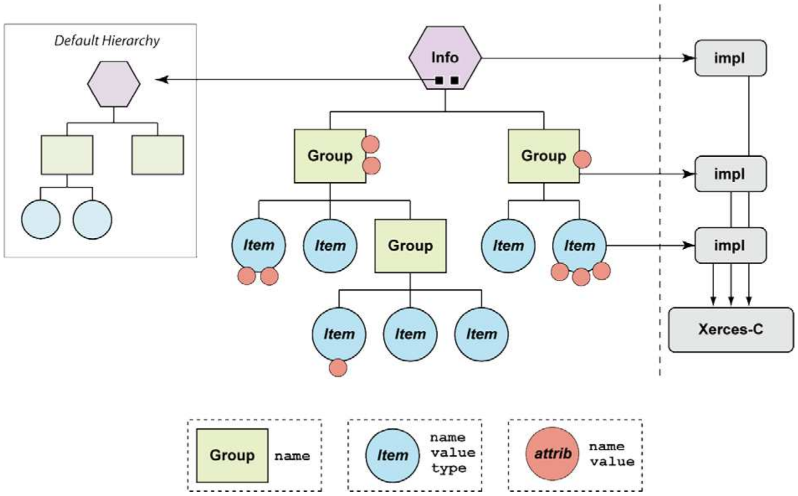


Fig. 5. Meta-data tree. The hierarchical info tree for meta-data storage. Groups and nodes are based on a thin layer implementation class which is build upon the Xerces library from the Apache project (see text for more detail).

2.9. Build system

The building process is organized with SCons, a python based, object oriented build system (http://www.scons.org). Since the build commands (such as create library, make executable, install file) are all available within Python scripts, customizing the building process is straightforward and not limited by pre-defined syntax, e.g. as for conventional **`make`** based builds. Several features should be mentioned:

2.9.1. Cross-platform handling

Extended from the cross-platform features already offered by SCons, the build process automatically detects the system, the compiler, the location of the required libraries and headers, sets required compilation flags, and so on. This is realized with a polymorphic base class in the build script that delegates platform specific tasks to concrete implementations; this is one of the advantages of having the build system within Python.

2.9.2. Stage

The build system mimics the installation of all necessary libraries, header, python scripts, and other files by placing them in a special stage directory in the IPLT development tree, from where they are visible to each other as if installed system wide. This staging allows each user/developer to have one or more fully functional, yet localized versions in unproblematic co-existence; and since the files in the stage directory are part of the SCons maintained dependencies, only the modified files are re-staged upon recompilation, leading to much faster compile/execute cycles than with traditional compile-then-install-all-then-run procedures. Ultimately, this features makes it very easy to rapidly prototype extensions to the existing code, and at the same time doing actual data processing.

2.9.3. Local extensions

The main build system makes several specialized, preconfigured build instances available, that can be imported into a local building script and thus integrate the local build with the overall IPLT one, in essence using the stage directory just described. The algorithm-containing directory of the IPLT source code may contain these local projects in own directories, which can even reside in their own source code repositories. During build, they will be automatically traversed and the build targets contained therein honored. Again, this feature aims at providing the developer with optimal independence and flexibility.

3. Processing capabilities

All the software engineering philosophy seems to be in vain if no experimental data is actually being processed by the package. This section gives an overview of the current processing capabilities, regarding both individual algorithms as well as our emerging electron crystallography processing module.

In the last 2 years, a collaboration within the European Union funded 3DEM Network of Excellence has focused on re-evaluating and re-implementing 2D electron crystallographic image processing, with IPLT as the underlying software framework. The requirements, constraints, and concepts emerging from this fruitful collaboration between the groups of H. Hebert (KI, Stockholm, Sweden), W. Kuehlbrandt (MPI, Frankfurt, Germany), J.M. Carazo (CSIC, Madrid, Spain) and A. Engel (MIH, Basel, Switzerland) have been the main driving force for the continued modification and extension of the IPLT interface and implementation. The scientific results from this collaboration will be described and discussed in due time by the involved parties, thus only a glance of what this collaboration has brought forth shall be mentioned here.

3.1. Currently implemented algorithms

All algorithms described here—unless otherwise indicated—can be applied to one, two, or three dimensional images. They are part of the currently available IPLT distribution.

Clear, Fill, Negate, Randomize: Setting image values to zero, a single value, its negative, or a random value, respectively.

Explicit Convolute: Convolution doing an explicit integral summation, rather than going via Fourier transforms.

Explicit Correlate: Cross Correlation doing an explicit integral summation, rather than going via Fourier transforms.

Discrete Fourier Transform: Fourier transform that honors the spatial and phase origin, respectively.

Fast Fourier Transform: Fourier transform that ignores the spatial origin and assumes physical image origin at $(0,0,0)$.

Histogram: Counts bin occupancy for a given range and bin count.

Normalize: Linear or logarithmic normalization to set values between zero and one.

Peak Search: Peak search routine, including various threshold values and exclusion areas.

Statistics: Calculates mean, standard deviation, minimum and maximum values.

FitGauss2D: Fits a 9-parameter Gaussian function to a 2D image: center (2 parameters), amplitude (1), major elliptical axes (2), ellipsis orientation angle (1), background plane (3).

Filters: Gaussian, Anisotropic, and Gaussian-Laplacian: Realspace, recursive filtering algorithms, based on the algorithm of (Young and van Vliet, 1995). Other filters (only for 2D images): Sharp-edge, Fermi, Butterworth.

Canny, *Sobel:* Edge detection (only for 2D images).

Erode, *Dilate*, *Open*, *Close*, *Hit-and-Miss:* Morphology based algorithms for binary images (only for 2D images).

Segment: Segmentation of an image into connected areas with the same pixel value (only for 2D images).

Threshold, *Density-Slice*: thresholding either with single or upper and lower limit.

Transformation: various transformations such as rotation, shifting in discrete or fractional pixel values, scaling, discrete shrinking.

3.2. The ex module

We have implemented the electron crystallography specific functionality in an IPLT module called *ex* (for *e*lectron *x*tallography), containing a collection of classes and procedures geared for this purpose: lattice search in a power spectrum or diffraction image, lattice refinement based on 2D Gaussian profile fitting, representation and interconversion of spatial and reciprocal unit cells, integration of diffraction peaks, tilt geometry encapsulation and determination, import/export of the CCP4 mtz file format to handle reflection data, and others. This is mainly used in a Python based 2D electron diffraction processing scheme (Fig. 6), complemented by a set of specific image viewer overlays, for instance a lattice overlay (Fig. 7) to facilitate interactive processing. A first version of this module is contained within the IPLT distribution.

3.3. Interaction with other packages

In addition, we have been working on providing optimal interaction with other, existing packages to utilize synergistic effects (Fig. 8). This is ongoing work for the MRC image processing package (Crowther et al., 1996) and the Xmipp software suit (Sorzano et al., 2004).

A Python wrapper has been designed for the MRC package that offers a similar concept and interface than the one found in IPLT: image files are wrapped as image handles, and the individual executables are addressed via algorithm objects that: (i) convert their stateful parameters to input parameters for the relevant executable; (ii) run the executable with the image file from the image handle; and (iii) parse the output and store the relevant values to be retrieved by the algorithm object interface. This interface is mostly identical to the IPLT Python interface.

In collaboration with the XMipp Developers (Sorzano et al., 2004), we have also started to implement a module for the interconversion of the C++ image representations of both packages. With the functionality exported to Python, a processing script may utilize the capabilities of both packages, using the native image implementation of each respective package, switching back and forth between them in a fast and efficient manner. In principal, each image processing package in the community that supports access via a python wrapper can be included in this scheme, given that such a conversion module is prepared. This would remove the tedious sharing of package functionality via image files.

4. Discussion

At the heart of an image processing class library two essential concepts must be present: (i) An encapsulation of an image; and (ii) routines that can manipulate such

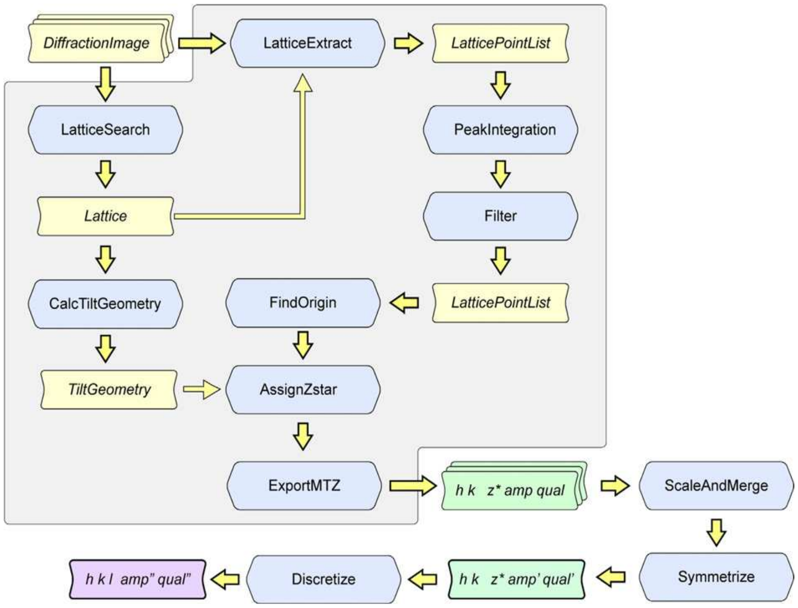


Fig. 6. 2D Electron diffraction processing scheme. To exemplify the implementation of image processing schemes using IPLT, this figure shows the 2D electron diffraction processing scheme as written in Python, based on algorithms (blue) and classes (yellow) from the IPLT *ex* module, which are written in either C++ or Python. The green and purple reflection lists are in the standard CCP4 reflection file format **`mtz`**, encapsulated with a **`ReflectionList`** class.

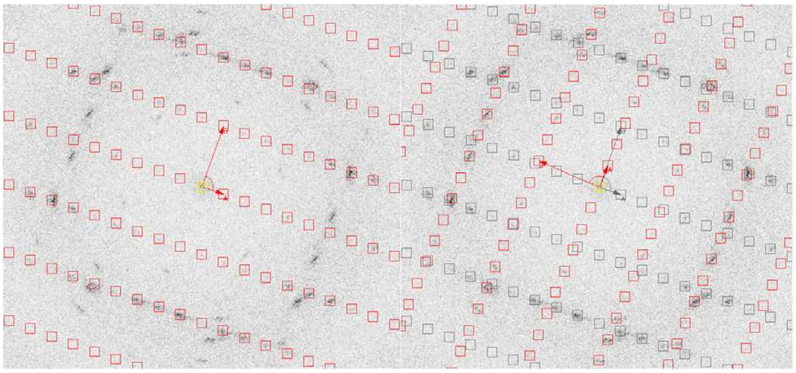


Fig. 7. Lattice overlay for 2D electron crystallography processing. A snapshot from one of our current projects illustrates the interactive image viewer with processing specific overlays. In this case, two lattice overlays are used to fit and mask the two lattices that are present. After fitting the first (left), a second overlay is added and becomes active (red) to fit the second lattice (right), while the first one remains fixed (grey).

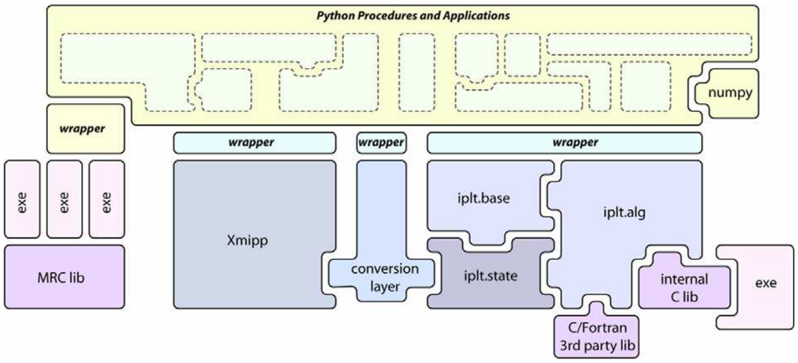


Fig. 8. IPLT in a wider context. The planned and currently developed image processing framework that is based on various components, unified by the Python language. Python functionality is either provided by C++ wrappers (in this case for Xmipp and IPLT), or Python wrappers emulating the IPLT interface (for MRC). The plethora of other Python modules is of course also available, such as Numpy (http://numeric.scipy.org).

an image. We have invested considerable effort in making sure that these two features are contained in IPLT in an efficient and flexible manner, for example by decoupling in-place and out-of-place application and implementation, or by offering high-level and low-level image access.

We also rationalize this more elaborate, seemingly complex design in view of our vision to design a framework that not only provides powerful image processing capabilities, but also is able to thrive in a dynamic, collaborative environment, within our own group as well as with other groups. We anticipate the recurrent needs of potential contributors, providing a system with maximal reusability yet relative ease of extension; for example by offering several interfaces, suitable for various levels of skill or interest. On the one hand an easy, error tolerant, but slower interface based on the image handle, which is sufficient for high-level manipulation and rapid prototyping, and on the other hand a more involving, error sensitive, yet fast interface for the performance critical algorithms. We are confident that these interfaces have evolved to a point where they are stable and no longer subject to re-design.

We have presented the current state of IPLT in a very technical, software engineering oriented manner, dwelling on interfaces and implementation and such. It goes without saying that the actual implementations of various algorithms are the bread and butter of image processing software, as these represent the methodological capabilities. Yet it is also in the nature of scientific research that algorithms are refined, methods mature, and novel approaches are taken. And it is within this context that the software architecture framework can help to facilitate the continuously evolving and changing methodology, by anticipating the dynamics within the team of contributors, by honoring that the prime interest of most coding scientists is the algorithm implementation, and not interface design, or maintainability, or code aesthetics.

It is evident that the dichotomy of end-user and programmer can no longer be upheld when considering the multitude of interfaces and their implementations; every user becomes a potential contributor. For scientific software, such as an image processing package, this is even more so, due to the different levels of expertise that each user brings along. We argue therefore that any scientific software package can only gain by offering interfaces on several levels, in order to allow a large diversity of contributions, even more so if these interfaces are part of an coherent and structured design. In a sense, IPLT is an experiment, a test to see if the initial strong emphasis on object oriented software design and the attempt of strict adherence to this concept is beneficial to the viability of this image processing package.

Acknowledgments

The authors thank Wanda Kukulski, Carlos Oscar Sanchez Sorzano, Roberto Marabini, Remco Wouts, Philipp Koeck, Björn Grundberg, Ronny Alvang and Johan Hebert for valuable feedback and/or contribution to IPLT. Financial support from the Maurice E. Müller Foundation, the NCCR Structural Biology, a Human Frontier Science Program long-term fellowship to S.B., and the European 3DEM NoE is gratefully acknowledged.

References

Beck, K., Andres, D., 2004. Extreme Programming Explained. Embrace Change. Addison-Wesley, Reading, MA.

Crowther, R.A., Henderson, R., Smith, J.M., 1996. MRC image processing programs. J. Struct. Biol. 116.

Fowler, M., 1999. Refactoring. Improving the Design of Existing Code. Addison-Wesley, Reading, MA.

Frigo, M., Johnson, S.G. 1998. FFTW: an adaptive software architecture for the FFT. In: Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing, vol. 3.

Josuttis, N.M., 1998. The C++ Standard Library: A Tutorial and Reference. Addison-Wesley, Reading, MA.

Philippsen, A., Schenk, A.D., Stahlberg, H., Engel, A., 2003. Iplt-image processing library and toolkit for the electron microscopy community. J. Struct. Biol. 144.

Sorzano, C.O.S., Marabini, R., Velázquez-Muriel, J., Bilbao-Castro, J.R., Scheres, S.H.W., Carazo, J.M., Pascual-Montano, A., 2004. XMIPP: a new generation of an open-source image processing package for electron microscopy. J. Struct. Biol. 148.

Tagari, M., Newman, R., Chagoyen, M., Carazo, J.M., Henrick, K., 2002. New electron microscopy database and deposition system. Trends Biochem. Sci. 27.

Young, I.T., van Vliet, L.J., 1995. Recursive implementation of the Gaussian filter. Signal Process. 44.

3.4 IPLT Publication 07 (2)

In the ongoing development of IPLT a paper summarizing the newest developments has been prepared. The main focus in this paper is directed on the implementations of algorithms and on the advanced GUI.

3.4.1 Contributions

As IPLT is a collaborative effort, many persons contributed to this paper. It contains algorithms I developed in the early times of IPLT (recursive filters, peak search and lattice determination). The Gauss fit used in the lattice determination and in the exact determination of the peaks in the cross correlation was implemented by Ansgar Philippsen. The origin determination was developed by Andreas Schenk. The algorithms were developed further by myself, Ansgar Philippsen and Andreas Schenk in a collaborative way. I contributed in the development of the GUI, where especially Andreas Schenk, Ansgar Phillipsen and Johan Hebert did a lot of prior work. As the first author I planed and wrote the paper.

3.4.2 The Paper

The paper will be submitted to the *Journal of Structural Biology* under the title "Versatile Image Processing of Electron Crystallography Data in IPLT". Following is the manuscript to be submitted.

Versatile Image Processing of Electron Crystallography Data in IPLT

G. A. Signorell[#], *A. Schenk*[#], *J. Hebert, V. Mariani, A. Engel, A. Philippsen*[*]

Maurice E. Müller Institute for Structural Biology, Klingelbergstr. 70, 4056 Basel, Switzerland

*Corresponding author

Ansgar Philippsen

`ansgar.philippsen@unibas.ch`

[#]These authors contributed equally to this work

Abstract

A sophisticated graphical user interface for IPLT is presented, which aims to make routine image processing with IPLT easier and faster, without sacrificing the flexibility and modularity of the framework. This novel user interface contains an improved python shell, an enhanced image viewer, a rapid method to design graphical one-button procedural processing schemes, and several algorithm related overlays. In addition, in-depth explanation is offered for selected image processing algorithms, such as automated peak search or lattice determination. Finally, the benefits of these new features are exemplified by calculating a projection map of the bacterial outer membrane protein NanC using three different processing schemes.

Introduction

The analysis of two dimensional crystals of membrane proteins using electron microscopy allows structural information of those proteins to be extracted; in the best cases, when well ordered, large, and single layered crystals are available, atomic resolution models can be obtained (Gonen et al., 2004; Mitsuoka et al., 1999; Henderson and Unwin, 1975; Unwin and Henderson, 1975). This 2D electron crystallographic methodology faces many hurdles, such as obtaining purified membrane protein, growth of suitable crystals, careful sample preparation and measurement, as well as extensive processing of the collected data.

The success of this data processing ultimately relies on the quality of the sample, yet it is facilitated by the use of sophisticated data processing algorithms. Therefore, the choice of algorithm type, order, and parameterization can and will have a large impact on quality of the result. In addition, the actual usage of these algorithms is based on their implementation in a software package, which means that the handling of the software also plays a role, albeit a lesser, in the successful processing of 2D electron crystallographic data.

Given the particular nature of this data, which usually consists of digitized micrographs or electron diffraction patterns, the employed processing algorithms are a mixture of generic as well as specialized image processing routines. The pioneering software package that offered a first set of combined image processing routines for the analysis of 2D crystallographic data was the MRC software package (Crowther et al., 1996), and it has established itself as the *de facto* standard in the field. Recently, a graphical front-end named 2DX has been developed (Gipson et al., 2006), which greatly circumvents the aged and rather cumbersome handling of the original package, and provides several extensions, such as a dedicated datafile to store all relevant parameters. Nevertheless, the originally intended usage and parameterization of the algorithms remains the same, and adding new methodologies to the MRC package is difficult.

Our group has been developing a novel open-source image processing framework named IPLT (*Image Processing Library & Toolbox*; `http://www.iplt.org`), primarily focused on 2D electron crystallography, with the goal to (i) re-evaluate and possible modify existing processing schemes, (ii) develop and test novel methods for accessing the high resolution information present in electron micrographs of 2D crystals, (iii) offer a software package that represents current concepts on software engineering; these goals and their implementations are described in [Philippsen et al., 2003] and [Philippsen et al., 2007]. In a nutshell, IPLT provides a modular, cross-platform architecture, with a core library and a large collection of image processing algorithms written in C++, a comprehensive set of Python (www.python.org) wrappers facilitated by boost-python (www.boost.org), and a flexible graphical user interface based on wxWidgets (www.wxwidgets.org). It is aimed at novice and expert users alike, providing several levels of possible contributions – the previous two papers clearly targeted interested developers, while this one is meant to demonstrate the highest and probably easiest level of usage via simple Python scripts and the graphical user interface.

In this paper we wish to demonstrate how we are using IPLT in our daily image processing routine, exemplified by describing the processing of 2D crystal data of the bacterial outer membrane protein NanC (Signorell et al, 2007). Particular focus is given to the new features of the (graphical) user interface, the interaction with the MRC software package, and technical excursion into some of the algorithms that have been incorporated.

Results & Discussion

Graphical User Interface Components

A considerable expansion of functionality has been added to the graphical user interface of IPLT - termed GIPLT - since the last publication (Philippsen et al, 2007), as will be described in detail below: The *python shell* not only provides sophisticated command line interaction with the framework, it also includes a straightforward way to build a graphical representation of a processing procedure. The *image viewer* has undergone a major face-lifting as well, with several embedded auxiliary windows providing additional information. Finally, the previously introduced concept of *overlays* is demonstrated with several algorithm related overlays.

Python Shell

The GIPLT *python shell* (see Figure 1A) provides the main access point for interactive work in IPLT. Via its command line, it provides a full Python interpreter environment, which is not only extended with the basic IPLT modules, but also with a specific graphical user interface module. This GUI module allows the creation and control of *image viewers*, *overlays*, and *procedures* (see below). Conversely, access to the Python interpreter is also possible within the C++ code of the GUI components.

Several noteworthy features enhance the usability of the *python shell* as a command line tool: (i) real-time adaptive command line completion is available via a search through the current interpreter namespace; (ii) history functionality allows recovering of preceding commands; (iii) python scripts can be interactively loaded and executed; (iv) typed input is logged and can be exported to a file for later reuse; (v) the output window features colour coding of input, standard output and error messages; (vi) the complete content of the output window is also logged and may be exported as well.

Graphical representation of python scripts

In order to facilitate standardized, linear processing schemes, a graphical *procedure* concept has been integrated into the *python shell*. As illustrated in Figure 1B, only a few Python commands are necessary to construct a simple *procedure* dialog box, where each button is associated with a Python function. Given the fact that the construction of this dialog box can be prepared within a Python script, which in turn can be run by the *python shell* upon startup of GIPLT, a processing environment can be generated which requires little or no usage of the command line interface and thus can be easily handled by novice users.

Image Viewer

The main and essential GIPLT element is the *image viewer*. As the name suggests, it is responsible for displaying images created or loaded with the *python shell*. Representative snapshots of the *image viewer* in action are given in Figures 2 and 3, whose elements are described below.

Versatile functionality is provided by numerous features: (i) memory efficient graphical implementation allows fast translation and zooming, independent of actual image size; (ii) numerical values are displayed on each pixel at a certain magnification scale; (iii) mapping of pixel values to greyscale can be adjusted either manually or based on an easily selected sub-region within an image; (iv) complex pixel values, for instance in Fourier transformed images, can be mapped to the HSV colour space, where the phase maps to the H, i.e. the colour hue, and the amplitude to V, i.e. the brightness value.

Auxiliary sub-windows complement the main image view with additional features. A *zoom* window gives a second view of the image at a different magnification – as with the main image view, numerical values are displayed on the pixels at a certain magnification level; this zoom window may serve as either a magnifying glass (Figure 2) or an overview (Figure 3). A *live-fft* displays an instant Fourier-transformation of an adjustable sub-region of the main image view, located around the mouse cursor (Figure 3). An *argand viewer* plots all pixel values of a selected sub-region as vectors in the complex plane (Figure 3). An *info panel* provides information about the mouse position and the current selection box (Figure 4). And finally, if at least one *overlay* is present, the *overlay manager* sub-window allows *overlays* to be activated, toggled on or off, and their parameters to be modified via overlay specific submenus (Figure 5).

Depending on the state of the displayed image, a default set of auxiliary sub-windows is opened: real-space images are accompanied with the *zoom* and *live-fft* sub-windows (Figure 2), Fourier-transformed images with the *zoom* and *argand* sub-windows (Figure 3).

While the *image viewer* window is an independent GUI element, it is accessible via the *python shell* command line as well, for example to retrieve a selected sub-region or to set a specific normalization range. This allows a fair amount of script based control, an important feature when building *procedures* (see above).

Overlay Concept

Overlays serve as a flexible extension to the image viewer to provide GUI elements for a multitude of tasks. Their concept was briefly introduced in (Philippsen et al, 2007): they allow graphical elements to be drawn on-top of a displayed image in the viewer, thus decoupling algorithm specific graphical display from the generic image one. They are constructed and assembled in a dynamic and interactive way via the *python shell*. They allow parameters to be set and retrieved via their python representation or via a graphical menu that is provided by the *overlay manager* sub-window in the *image viewer* (see above).

Several *overlays* can be added to a single *image viewer* as illustrated in Figure 4; to avoid graphical cluttering and ambiguous interaction with the mouse and keyboard, only one *overlay* is active at a time and receives mouse and keyboard input. In addition, the *image viewer* will only route mouse and keyboard events to the active *overlay* if the <CTRL> key is pressed to ensure that basic operations like moving or zooming the image itself are never affected by an *overlay*.

CTF Overlay

The *CTF overlay* is used for interactive determination of the image defocus by fitting the zero lines of a simulated (anisotropic) contrast transfer function (CTF), displayed as concentric ellipses, into the Thon rings of a Fourier transformed image, as demonstrated in Figure 4. As an option, only one half of the ellipses can be displayed to allow better visibility of the Thon rings. Also, the aspect ratio of the ellipsis can be kept fixed or be varied freely to account for astigmatism.

Once the defocus is determined, the defocus values and the elliptical rotation angle can be retrieved from the overlay via the *python shell*, and in turn used to correct the optical artefact introduced by the CTF (as shown below for the NanC processing).

Lattice Overlay

A typical snapshot of a *lattice overlay* is given in Figure 3; it provides interactive fitting of a 2D lattice, which is defined by an origin, two lattice vectors, and a barrel- as well as a spiral-distortion (see Eq.(2) in Materials and Methods). In addition, if the unit cell parameters are known, the tilt geometry is displayed as well, and adjusted automatically upon modification of the lattice.

Each lattice point can be dragged around individually with the mouse, upon which the lattice automatically adapt by continuously solving Eq.(2) based on the new position of the shifted lattice point. This adaptation can be controlled by fixing one or two lattice points, or none at all, where the following restrictions apply: (i) if no lattice point is fixed, only the length of the two vectors will be modified, the origin and the angle between the vectors remains unchanged; (ii) if one lattice point is fixed, the origin will also remain unchanged, but both vectors adjust freely; (iii) if two lattice points are fixed, the origin will also change, together with the two vectors.

The lattice as a whole can be shifted, resulting in a change of the origin. Also, the two lattice vectors can be rotated around the origin. It must be mentioned that during lattice modification, the original lattice is displayed as a reference, and explicit acceptance or rejection of the new lattice is required to end the modification process.

While the individual adjustment of lattice points provides sophisticated tweaking of the lattice parameters, it is still somewhat tedious to manually place a point precisely on the centre of a peak in the image. To this end, the *lattice overlay* provides automatic peak fitting of the lattice point under adjustment: with a single key press, a sub-region around the current lattice point position is subjected to a 2D Gaussian fit, as described in the Materials and Methods section, and, if the centre of this fit lies within the sub-region, it is set as the new coordinate of the lattice point, with subsequent adaptation of the complete lattice as described above. It should be noted that this not only greatly increases the speed of manual fitting, it also increases the precision, since the 2D Gaussian fit provides sub-pixel precision of the peak position.

The resulting lattice can be retrieved from the *lattice overlay* via the *python shell* in form of a standard IPLT `Lattice` object, which is used in numerous other places, such as automatic lattice determination or lattice refinement (as described later).

Pointlist Overlay

As the name suggests, the *PointList overlay* allows the display of a list of points on top of an image, and is usually employed to visualize the results of a peak search, as exemplified in Figure 5.

Spotlist Overlay

The *Spotlist overlay* is primarily aimed at the interaction between IPLT and MRC. It displays a static, non-editable lattice, and allows individual lattice points to be selected and de-selected. The list of selected points can then be saved into an MRC compatible file. The import of such a file is also possible.

Unit Cell Overlay

The *UnitCell overlay* draws a graphical representation of a unit cell on top of an image, automatically choosing real-space or reciprocal-space unit cell vectors. An example is given in Figure 2.

Algorithms

Peak search

Automatic analysis of measured or calculated diffraction patterns must necessarily include a way to identity the reflection peaks; the same is true for images analysed with auto- or cross-correlation methods, where the signal peaks must be located. A flexible and powerful peak search algorithm has been implemented in IPLT for this purpose, whose core algorithm is explained in Figure 5A. Its main feature is a search within a localized window around a candidate pixel, with a threshold relative to the standard deviation of the image. The introduction of an inner and outer window allows tweaking according to expected peak spread and better differentiation of signal versus background, resulting in a reduction of false positive peaks caused by noise.

Additional features of this algorithm include the specification of exclusion regions, minimal distance between peaks, and support for 1D, 2D as well as 3D peak search. Since this algorithm is implemented in C++ and reflected into Python, it can easily be called from the shell, included in Python procedures, or incorporated in other C++ based algorithms, as done for the automatic lattice determination algorithm, explained below.

Automated Lattice Determination and Refinement

The crystalline nature of 2D electron crystallography samples gives rise to reflection peaks in Fourier space, which are located on reciprocal lattice points. For the case of crystal data with only one lattice, IPLT provides fully automated algorithms to (i) identify the lattice vectors (ii) find the origin in diffraction images where a part of the diffraction pattern is obstructed by a beam-stop, in particular the origin peak; (iii) refine all lattice parameters based on a number of peaks. The result of this automated lattice determination is easily available through a graphical lattice overlay (see above).

The inner workings of these algorithms are explained in the Materials and Methods section; in a nutshell, the difference vectors between diffraction peaks are used to extract the lattice vectors, and a description of a lattice including barrel and spiral distortion is used for the refinement.

MRC Python Interface

Although this is by no means part of the normal IPLT package, it must be mentioned that we have designed a sophisticated set of python wrappers to access the collection of MRC executables, as described previously (Philippsen et al, 2007). This allows integration of MRC routines into the context of IPLT scripts and interactive sessions; this mainly serves to ensure a transition period in which the daily image processing routine can continue while algorithms and user interface add-ons are being (re-)implemented in IPLT to replace their MRC counterparts.

Exemplified Walkthroughs

To illustrate how the previously described algorithms and graphical user interface components can be used in routine image processing of 2D crystallographic data in IPLT, this section presents three processing schemes to obtain a 2D projection map: (i) simple Fourier filtering with subsequent correlation averaging (after Saxton and Baumeister, 1982); (ii) enhanced processing with CTF correction and peak refinement; (iii) combined IPLT-MRC processing, including unbending, as described in [Signorell et al, 2007]. The results of these processing schemes are presented in Figure 6.

The data used herein is part of a dataset from tubular NanC crystals, described in more detail in Signorell et al, 2007. All scripts and images from this section are available online in the tutorial section of the IPLT homepage at `http://www.iplt.org`.

Fourier Filtering and Correlation Averaging

In the *python shell*, a scanned raw image is loaded into IPLT and Fourier transformed with FFT. The resulting image is displayed with an *image viewer*. Due to the fact that two lattices are present (from the flattening of the tubular crystal), the lattice cannot be determined with the automatic lattice determination algorithm, and must be fitted manually. To this end, a *lattice overlay* is added to the *image viewer* and fitting may proceed (Figure 3). Once the lattice parameters are determined, a simple masking algorithm removes data outside the lattice point regions, and an inverse Fourier transformation generates a filtered version of the raw image.

The filtered image is now opened in a second *image viewer* (Figure 2), and a manual selection of a region - at about twice the size of a unit cell and with well defined substructure - is utilized to generate a reference, which in turn is cross-correlated with the original image. This cross-correlation image is fed into the peak search algorithm, and the resulting peak list is then used to extract sub-regions of the original image, which are averaged to produce the final projection map, shown in Figure 6A.

Enhanced Processing with CTF Correction and Peak Refinement

An enhanced version of the above processing scheme is described next. As a first step, a *CTF overlay* is added to the display of the Fourier transformed image. The resulting defocus is utilized in the Wiener Filter algorithm (see Material and Methods) to correct the image for the optical artefact introduced by the CTF. As next steps, the Fourier filtering, the cross-correlation with the reference, and the peak-search are done as described above, with the CTF corrected image. The cross-correlation peaks are individually subjected to 2D Gaussian fitting (see Materials and Methods), resulting in refined, sub-pixel precise positions. The sub-pixel position shift is taken into account by applying a corresponding phase shift to the extracted sub-images in Fourier space, instead of performing a more complicated translation in real-space. As the last step, the sub-images averaged.

To summarize, the Fourier filtering and cross-correlation processing scheme has been ameliorated with CTF correction and sub-pixel position accuracy of the averaged unit cells, resulting in an improved projection map, as displayed in Figure 6B.

Semi Automatic Processing using IPLT and MRC (including unbending)

The above scheme is further improved by correction of crystal distortions, performed with the unbending procedures in MRC (Henderson et al, 1986). A detailed explanation and discussion of this scheme is given in [Signorell at al, 2007]; in the context of this work, the following graphical user interface elements were instrumental in providing a seamless integration of IPLT with MRC: (i) determination of both lattices with two *lattice overlays*, where the second lattice is given to MRC to exclude spots which are in too close proximity (ii) determination of the initial defocus with the *CTF overlay*; (iii) generation of the spot list, used for creating the first unbending reference in MRC, by means of the *Spotlist overlay*;

Interaction with MRC can be directly provided with the MRC Python wrapper (see above), by calling MRC executables from within the IPLT scripts. Alternatively, values can be saved to and retrieved from MRC compatible files, such as the spotlist or the 2DX/MRC specific datafile, with a subsequent call of MRC executables in a stand-alone fashion.

The resulting projection map based on this unbending enhanced processing scheme is shown in Figure 6C. It should be noted that the final projection map of NanC in [Signorell et al, 2007] contains merged data from several images, as well as inclusion of crystal symmetry. These steps were done with MRC and are omitted here, to allow easier comparison of the projection maps from all three schemes.

Acknowledgment

This work was supported by the Maurice E. Müller Foundation of Switzerland, by the Swiss National Foundation (SNF) within the framework of the National Center of Competence in Research for Structural Biology, the NoE 3D-EM (EU project LSHG-CT-2004-502828), and the HT-3DEM (EU project LSHG-CT-2005-018811).

Materials and Methods

Electron Microscopy

The NanC crystals were produced and the images acquired with a Phillips CM 200 FEG cryo electron microscope as described in (Signorell et al, 2007). The negatives recorded on Kodak SO163 film were scanned at 2000 lines/cm on a Heidelberg Primescan D 7100, resulting in a resolution of 1 Å/pixel at the sample level.

Algorithmic Implementations

Recursive filters in real space

Gaussian or anisotropic filters can be implemented by calculation of a convolution in real space. Since the computation time of algorithms is dependent on the width of the Gaussian kernel (σ), they are less efficient at higher filter strength. Gaussian or anisotropic filters are thus often implemented in Fourier space to better meet the speed requirements for the processing of large images. The main problem of this implementation is the expensive Fourier transform and the back transform. The recursive implementation in real space of the presented filters avoids performing these expensive steps. The value of σ does not influence the speed of the algorithm (the number of multiplications and additions is constant), and thus this recursive implementation is in general claimed to be faster than the FFT implementation or a repeated convolution with a kernel (Young and van Vliet, 1995). A recursive implementation of a Gaussian and an anisotropic (edge preserving) filter was developed in C++ using the ideas presented in (Young and van Vliet, 1995).

2D Gaussian Fit

A 2D Gaussian function based on 9 parameters can be fitted to an image using a non-linear least squares fitting algorithm, where each pixel at (x,y) is supposed to satisfy

$$G(x,y) = A\exp\left[\left(\frac{x'\cos\omega - y'\sin\omega}{B_x}\right)^2 + \left(\frac{y'\cos\omega + x'\sin\omega}{B_y}\right)^2\right] + P_x x' + P_y y' + C\,, \qquad (1)$$

with $x' = x - x_0, y' = y - y_0$, and the parameters are centre (x_0,y_0), amplitude (A), major elliptical axes (B_x, B_y), ellipsis rotation angle (ω) and background plane (P_x,P_y,C). A second version of this algorithm allows an explicit list of (x,y) coordinates and their corresponding values to be fitted. As is explained in the Results and Discussion section, this algorithm has proven to be especially useful in determining the centre of reflection peaks at fractional pixel position, or background determination in the area of the reflection peak.

Lattice Determination and Refinement

The position of each lattice point is based on its index (h,k) and is given by $\mathbf{L} = \mathbf{O} + h\mathbf{P} + k\mathbf{Q}$, where **O** is the origin and **P**,**Q** are the two reciprocal lattice vectors. A more complex model includes barrel and spiral distortion, defined by two constants *Kb* and *Ks*, and is given by

$$\mathbf{L} = \mathbf{O} + \begin{pmatrix} 1 + K_b\left|h\mathbf{P} + k\mathbf{Q}\right|^2 & K_s\left|h\mathbf{P} + k\mathbf{Q}\right|^2 \\ -K_s\left|h\mathbf{P} + k\mathbf{Q}\right|^2 & 1 + K_b\left|h\mathbf{P} + k\mathbf{Q}\right|^2 \end{pmatrix}\left(h\mathbf{P} + k\mathbf{Q}\right) \qquad (2)$$

The *lattice search algorithm* works in the following way:

In a first step a peak search is performed and the parameters (sensitivity and desired peak size) of the peak search algorithms are refined according to the number of peaks found. This is achieved to avoid too many peaks (likely to introduce more noise peaks) or too few peaks. From this peak list, the distance vector between each pair of peaks is calculated. All these distance vectors contribute to a new image, the *vector image*. The *vector image* is generated starting from an image with all the pixel values set to zero. The pixel value is increased by one at each position of the image (index) defined by the components of the distance vectors. On this image a second peak search is performed and a second *vector image* is generated from these peaks. In this second *vector image* the two closest points to the centre that are not collinear are taken as a first guess for the lattice vectors.

The found lattice is now superimposed on the original image. At each of the predicted lattice points, a 2D Gaussian function (see above) is fitted to determine the exact position of the peak. This list of positions is fed into the *lattice refinement algorithm*, which determines the best set of lattice parameters for Eq.(2). If both barrel- and spiral-distortions are fixed to zero, a linear-least squares routine is used, otherwise a more involving non-linear least squares routine must be employed. Both variants are implemented using a Levenberg-Marquardt algorithm from the Gnu Scientific Library (http://www.gnu.org/software/gsl).

The *lattice refinement algorithm* can also be called stand-alone outside the context of the *lattice search algorithm*.

Origin Determination in Diffraction Patterns

The origin of a diffraction pattern is determined by two heuristic algorithms, (i) by a global 2D Gaussian profile fit over the background, or (ii) by a convolution with the mirror image and subsequent maximum extraction.

Wiener Filter

Correction of the CTF is done with a Wiener filter, implemented based on the description in [Reimer 1997], p.258, Eq.(6.70)

Graphical User Interface

The graphical user interface (GUI) was implemented using the wxWidgets library (http://www.wxwidgets.org), providing a native look and feel as well a cross-platform compatibility. Subwindows of the shell and the DataViewer were realised using the wx advanced user interface (wxAUI) which provides dockable panes.

References

Crowther, R. A., Henderson, R. and Smith, J.M., 1996. MRC image processing programs. J Struct Biol, 116(1):9-16.

Gipson, B., Zeng, X., Zhang, Z.Y., and Stahlberg, H., 2007. 2dx-User-friendly image processing for 2D crystals. J Struct Biol. 157(1)

Gonen T., Sliz P., Kistler J., Cheng Y. and Walz T., 2004. Aquaporin-0 membrane junctions reveal the structure of a closed water pore. Nature 429(6988):193-7.

Henderson, R. and Unwin, P.N., 1975. Three-dimensional model of purple membrane obtained by electron microscopy. Nature, 257(5521):28-32.

Henderson, R., Baldwin, J. M., Downing, K. H., Lepault, J. and Zemlin, F., 1986. Structure of purple membrane from halobacterium halobium: recording, measurement and evaluation of electron micrographs at 3.5 Å resolution Ultramicroscopy, 19(2): 147-178

Mitsuoka, K., Murata, K., Walz, T., Hirai, T., Agre, P., Heymann, J.P., Engel, A. and Fujiyoshi, Y., 1999. The structure of aquaporin-1 at 4.5-A resolution reveals short alpha-helices in the center of the monomer. J Struct Biol, 128(1):34-43.

Philippsen, A., Schenk, A.D., Stahlberg, H. and Engel, A., 2003. Iplt--image processing library and toolkit for the electron microscopy community. J Struct Biol, 144(1-2):4-12.

Philippsen, A., Schenk, A.D., Signorell, G.A., Mariani, V., Berneche, S. and Engel, A., 2007. Collaborative EM image processing with the IPLT image processing library and toolbox. J Struct Biol. 157(1)

Reimer L., 1997. Transmission Electron Microscopy, Springer, Berlin

Saxton, W.O. and Baumeister, W., 1982. The correlation averaging of a regularly arranged bacterial cell envelope protein. J Microsc, 127(Pt 2):127-38.

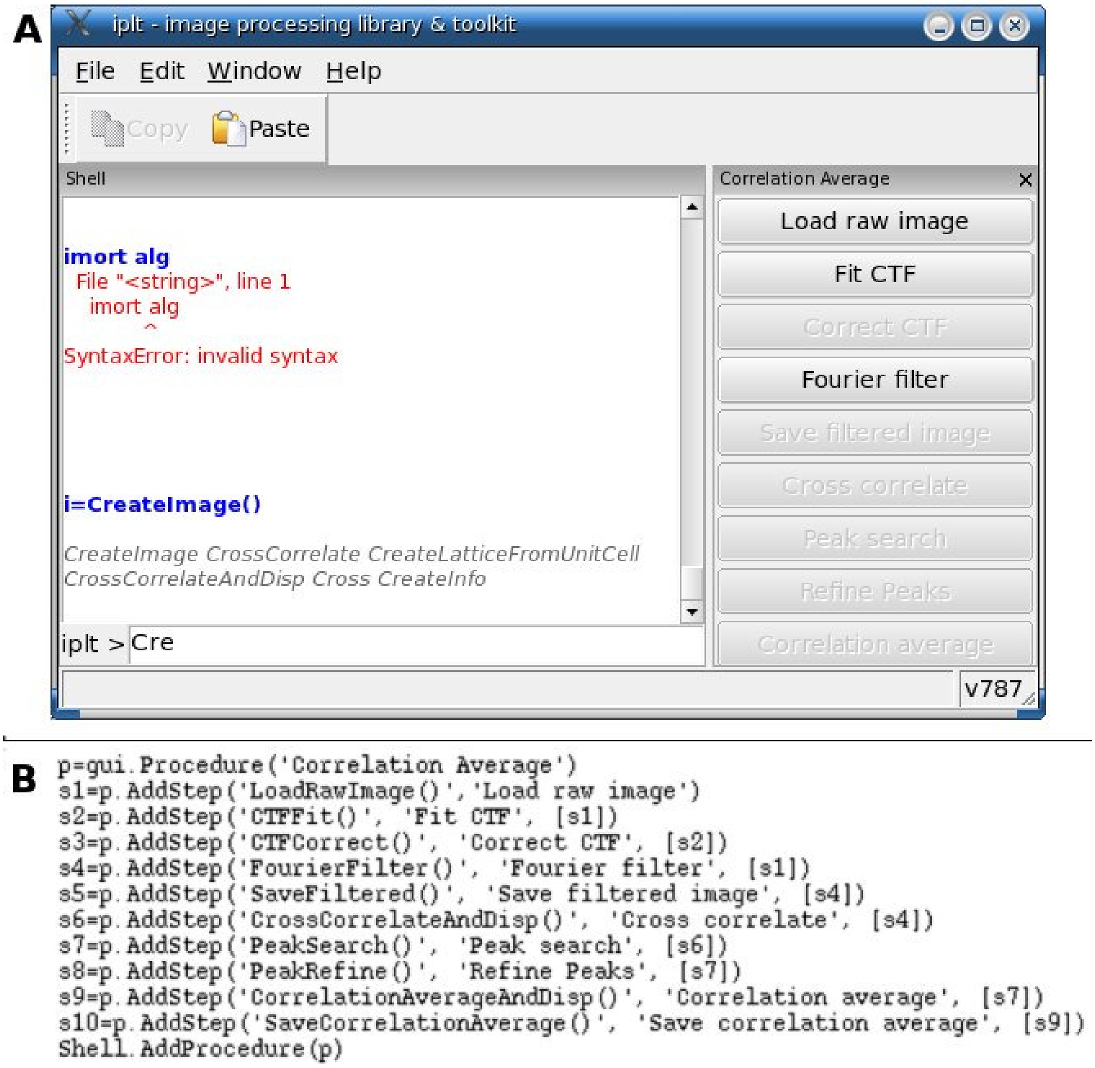
Shaw, P.J. and Hills, G.J., 1981. Tilted specimen in the electron microscope: A simple specimen holder and the calculation of tilt angles for crystalline specimens, Micron 12, 279-282.

Signorell, G.A, Remigy, H.W., Chami, M., Schenk, A.D., Philippsen and A., Engel, A., 2007. Projection Maps of Three Outer Membrane Proteins, Submitted

Unwin, P.N. and Henderson, R. 1975. Molecular structure determination by electron microscopy of unstained crystalline specimens. J Mol Biol, 94(3):425-40.

Young, I.T., van Vliet, L.J., 1995. Recursive implementation of the Gaussian filter. Signal Process. 44.

Zampighi, G.A., Kreman, M., Lanzavecchia, S., Turk, E., Eskandari, S., Zampighi, L. and Wright, E.M., 2003. Structure of functional single AQP0 channels in phospholipids membranes. J Mol Biol, 325(1):201-10.

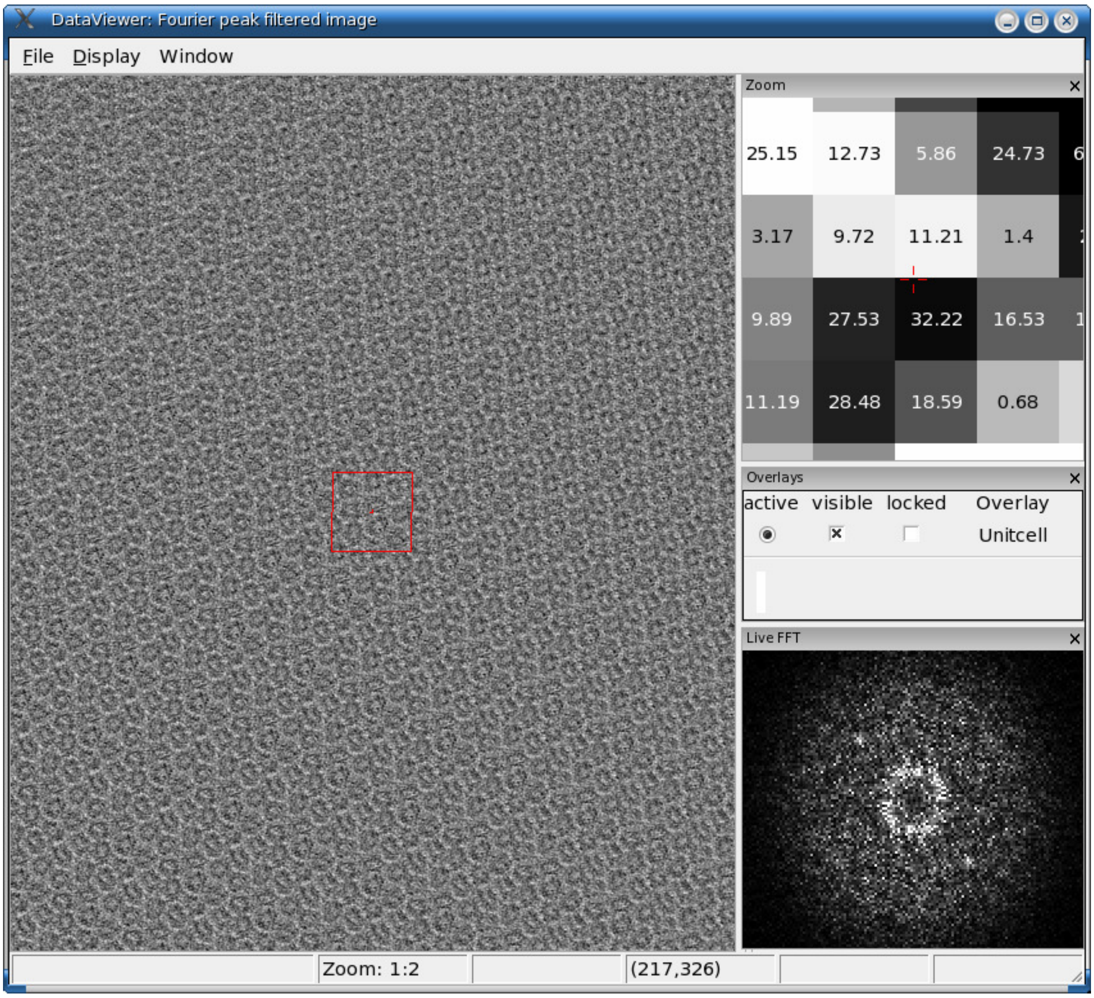


```
imort alg
  File "<string>", line 1
    imort alg
        ^
SyntaxError: invalid syntax



i=CreateImage()

CreateImage CrossCorrelate CreateLatticeFromUnitCell
CrossCorrelateAndDisp Cross CreateInfo
iplt > Cre
```

```
p=gui.Procedure('Correlation Average')
s1=p.AddStep('LoadRawImage()','Load raw image')
s2=p.AddStep('CTFFit()', 'Fit CTF', [s1])
s3=p.AddStep('CTFCorrect()', 'Correct CTF', [s2])
s4=p.AddStep('FourierFilter()', 'Fourier filter', [s1])
s5=p.AddStep('SaveFiltered()', 'Save filtered image', [s4])
s6=p.AddStep('CrossCorrelateAndDisp()', 'Cross correlate', [s4])
s7=p.AddStep('PeakSearch()', 'Peak search', [s6])
s8=p.AddStep('PeakRefine()', 'Refine Peaks', [s7])
s9=p.AddStep('CorrelationAverageAndDisp()', 'Correlation average', [s7])
s10=p.AddStep('SaveCorrelationAverage()', 'Save correlation average', [s9])
Shell.AddProcedure(p)
```

Figure 1

(A) Snapshot of the GIPLT *python shell* in action, including a graphical *procedure* on the right hand side. Input from the command prompt is displayed in blue, normal output in black, error messages in red, and suggestions for the command line completion in grey. (B) The essential code used to create the shown *procedure*; upon creation of a **`gui.Procedure`** object, individual steps are added to it, which contain the name of the Python function to be called, the name of the button, and optionally a list of dependencies, i.e. the steps that need to be completed prior to activating the button. In this snapshot, step **`s1`** ("Load raw image") has been done, therefore the two buttons "Fit CTF" and "Fourier Filer" have become active, the others remain inactive.



25.15	12.73	5.86	24.73	6
3.17	9.72	11.21	1.4	
9.89	27.53	32.22	16.53	1
11.19	28.48	18.59	0.68	

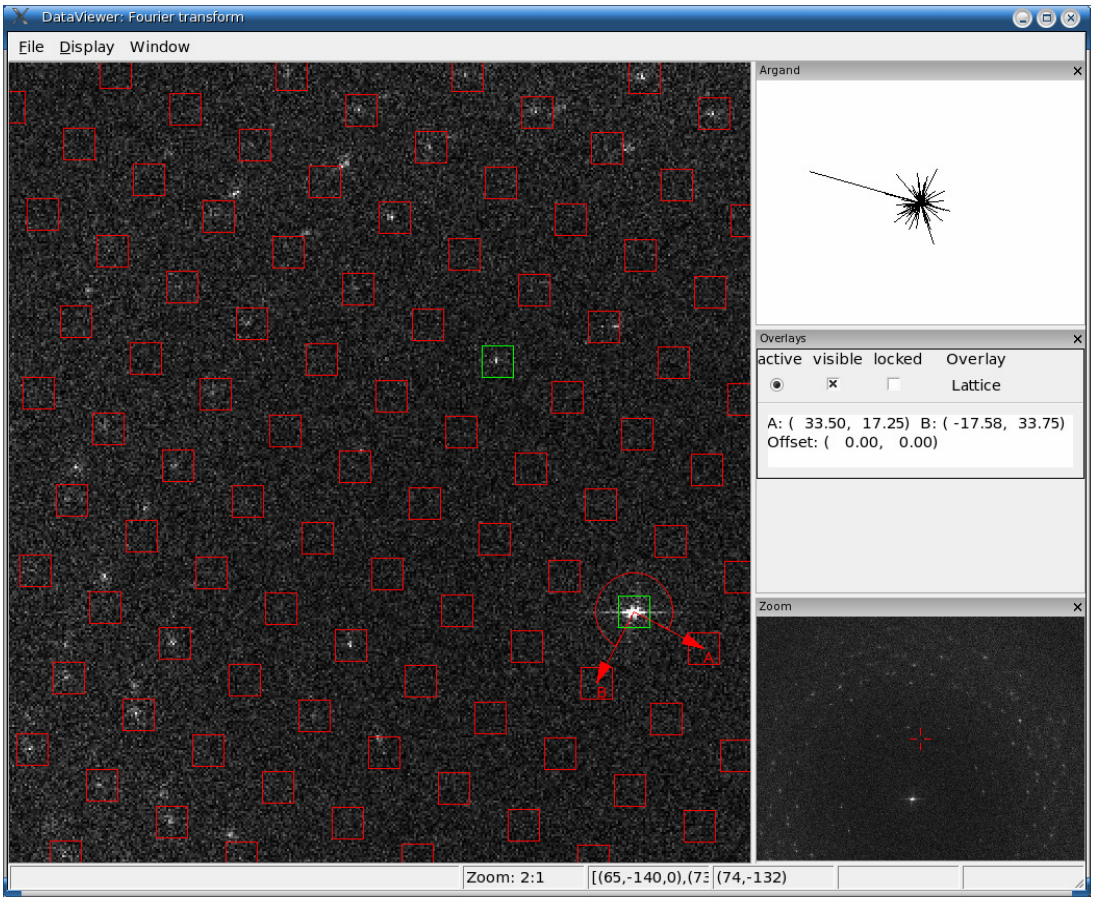
active	visible	locked	Overlay
●	☒	☐	Unitcell

Zoom: 1:2

(217,326)

Figure 2

Snapshot of the GIPLT *image viewer* displaying a real-space image (the Fourier filtered image from the exemplified walkthrough). The *zoom* sub-window at the top right is at a high enough magnification to display the numeric pixel values around the position of the mouse pointer. The *live-fft* sub-window is located at the bottom right and is updated upon pressing the <SHIFT> key during mouse movement. The red square in the main view is drawn by the *UnitCell overlay*, which is accessible in the *overlay manager* sub-window on the centre-right.

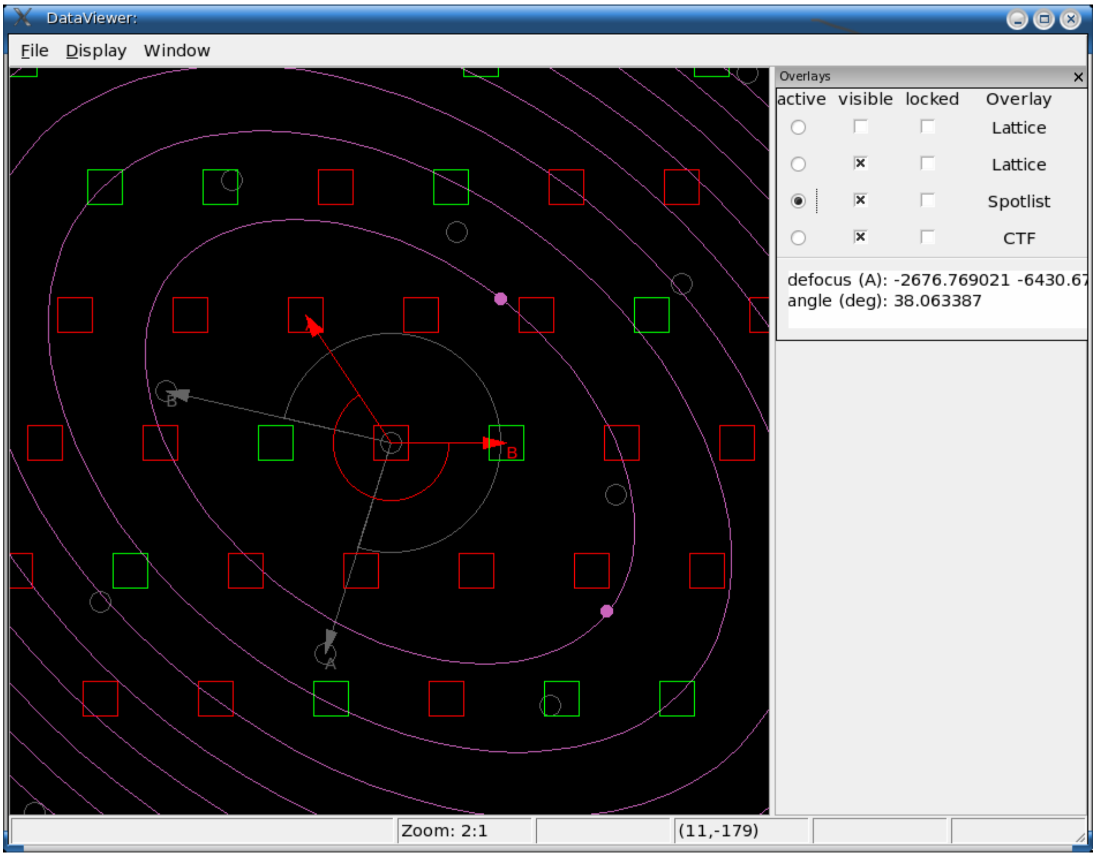


active	visible	locked	Overlay
●	☒	☐	Lattice

A: (33.50, 17.25) B: (-17.58, 33.75)
Offset: (0.00, 0.00)

Figure 3

Snapshot of the GIPLT image viewer displaying a Fourier transformed image (FFT transformed raw image from exemplified walkthrough). On the top right, the *argand* sub-window is displaying the complex pixel values for a selected sub-region (not shown). A *zoom* sub-window on the bottom right is utilized as an overview. The graphical elements on top of the main image are drawn by a *lattice overlay*, which is accessible via the *overlay manager* sub-window at the centre right. In this case, the two green boxes represent fixed lattice point locations, and dragging any other lattice point will not modify these (see text for more detail).

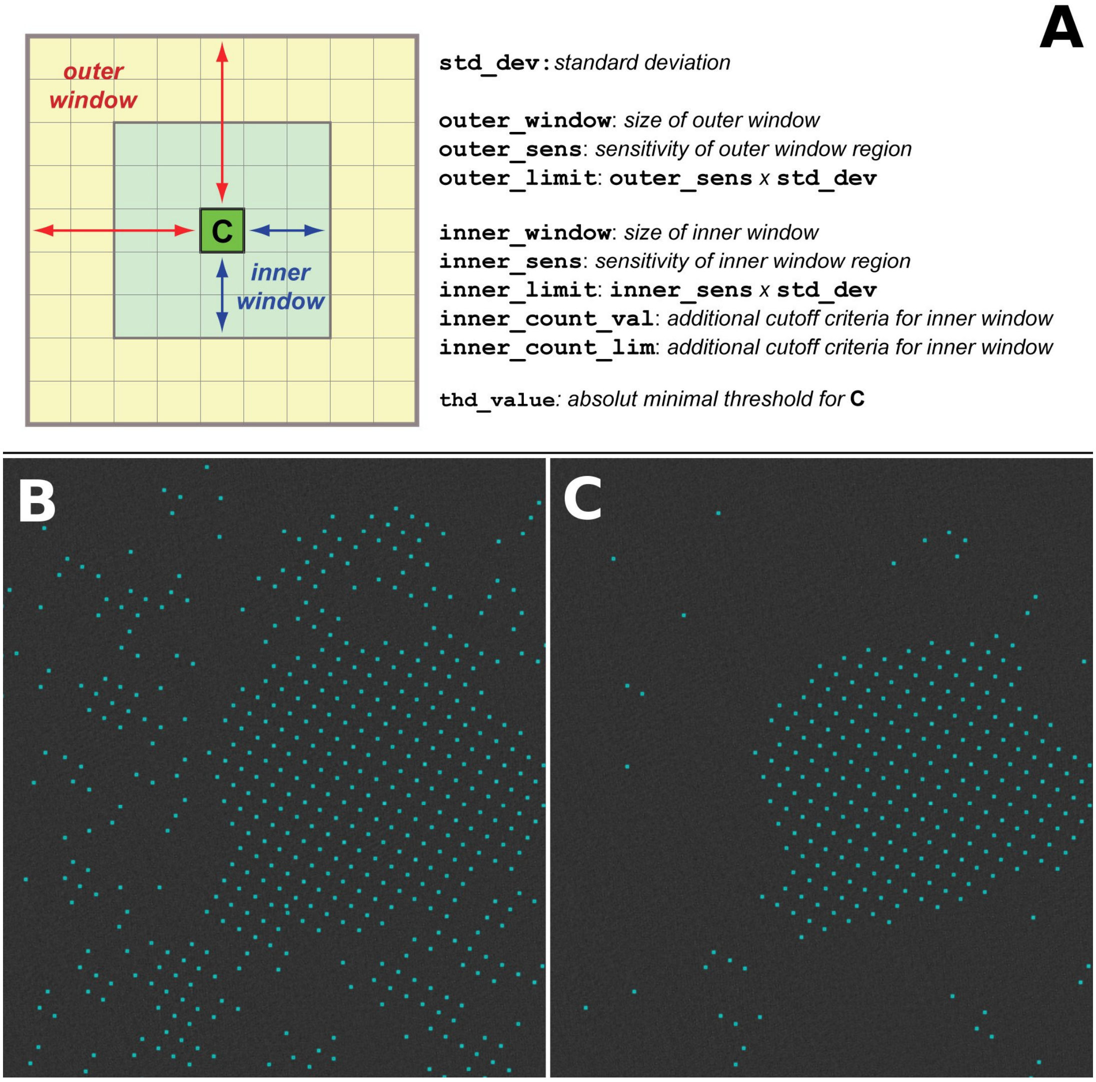


active	visible	locked	Overlay
○	☐	☐	Lattice
○	☒	☐	Lattice
●	☒	☐	Spotlist
○	☒	☐	CTF

defocus (A): -2676.769021 -6430.67
angle (deg): 38.063387

Figure 4

Illustrative collection of the *overlays* mentioned in the text, demonstrating how combination of *overlays* can be used to assemble a custom graphical user interface on the fly. The *overlay manager* sub-window at the top right corner is used for activation, (de)activation and locking of the individual *overlays*; in addition, the rightmost column with the name of each *overlay* provides a submenu, where parameters can be set (not shown).



`std_dev`: *standard deviation*

`outer_window`: *size of outer window*
`outer_sens`: *sensitivity of outer window region*
`outer_limit`: `outer_sens` *x* `std_dev`

`inner_window`: *size of inner window*
`inner_sens`: *sensitivity of inner window region*
`inner_limit`: `inner_sens` *x* `std_dev`
`inner_count_val`: *additional cutoff criteria for inner window*
`inner_count_lim`: *additional cutoff criteria for inner window*

`thd_value`: *absolut minimal threshold for* **C**

Figure 5

The peak search algorithm is illustrated in (A); a candidate pixel **C** will be accepted as a peak under the following conditions: (i) its value must be larger than `thd_value`; (ii) its value must be `outer_limit` larger than all values of the outer window pixels; (iii) its value must be `inner_limit` larger than all values of the inner window pixels; (iv) there must be at least `inner_count_lim` pixels in the inner window that have a value larger than `inner_count_val`. Note that the outer window does not include the inner window region.

Results of a peak search on a cross-correlation image with different sensitivity are given in (B) and (C), displayed using the *Peaklist overlay*. The outer window sensitivity relative to the standard deviation is 1.0 in (A) and 2.0 in (B); the inner window size is zero.

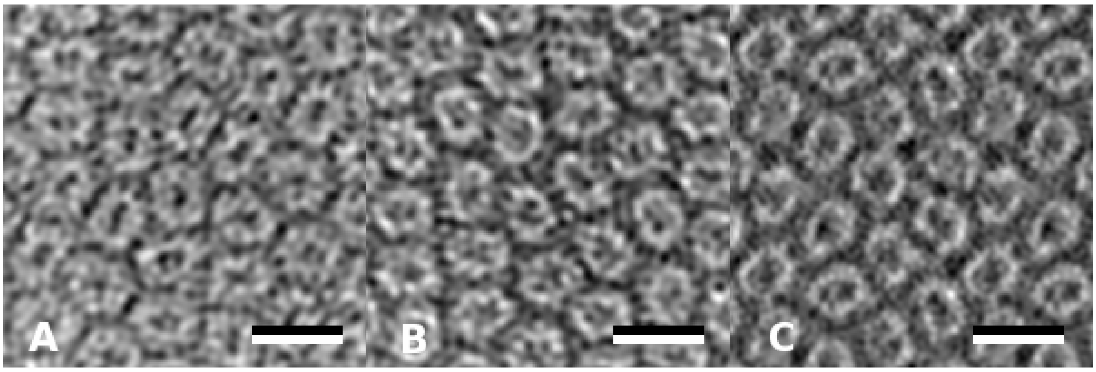


Figure 6

The scale bars on the images represent 50 Å. (A) Resulting projection map after Fourier filtering and correlation averaging. The pores are clearly visible as rings. (B) Resulting projection map after the inclusion of a CTF correction and a peak refinement. The noise is reduced and the pores are better visible. (C) Projection map of one image using the combined IPLT/MRC algorithms. Refer to the text for more details.

3.4.3 Ongoing Development

The development of IPLT is continued in a number of areas: Simulations of the TCIF (Valerio Mariani), the expansion of the GUI (Andreas Schenk), the automated handling of multiple lattices (me) and the diffraction processing are the "hot"-topics at the moment.

3.5 Image Processing of KdgM, KdgN and NanC

The image processing of KdgM, NanC and KdgN is a good example for our development in the image processing. KdgM was first processed on a SGI machine, using XIMDISP, an MRC program [9] for the indexing of the two lattices. During the data collection and image processing of KdgM, the port of MRC to Linux operating systems advanced and the GUI of IPLT was developed in such a way that I could make initial processing trials with the new software and compare the results on both systems. The GUI of IPLT proved to be a great improvement in the indexing of the difficult lattices of KdgM as the crystals were double layers and the two lattices are arranged in a unfavorable way (see section 2.3.6 for details).

Crystals of a suitable quality for image processing of NanC and KdgN were available at a later time and so this new projects were set up and computed completely on a Linux machine. In this phase of the project, there were still things to improve and the processing of this two proteins helped to find the weak spots and to improve the whole system.

A full description of the image processing is given in the paper on the three porins described in section 2.3.6.

3.6 Image Processing of HasA/R

Data Collection and Indexing Data collection in the HasA HasR project focuses mainly on the collection of diffraction data. Nevertheless a set of images of single layered crystals were recorded at the cryo electron microscope by Mohamed Chami. The images were taken at a magnification of 50’000 and were then scanned at 2000 lines/cm, resulting in a sampling of 1 Å/pixel.

The images were of a very high quality and the indexing (see figure 3.6) was straight forward. The unit cell was determined to have the parameters a = 126 Å, b = 101 Å andγ = 90°.

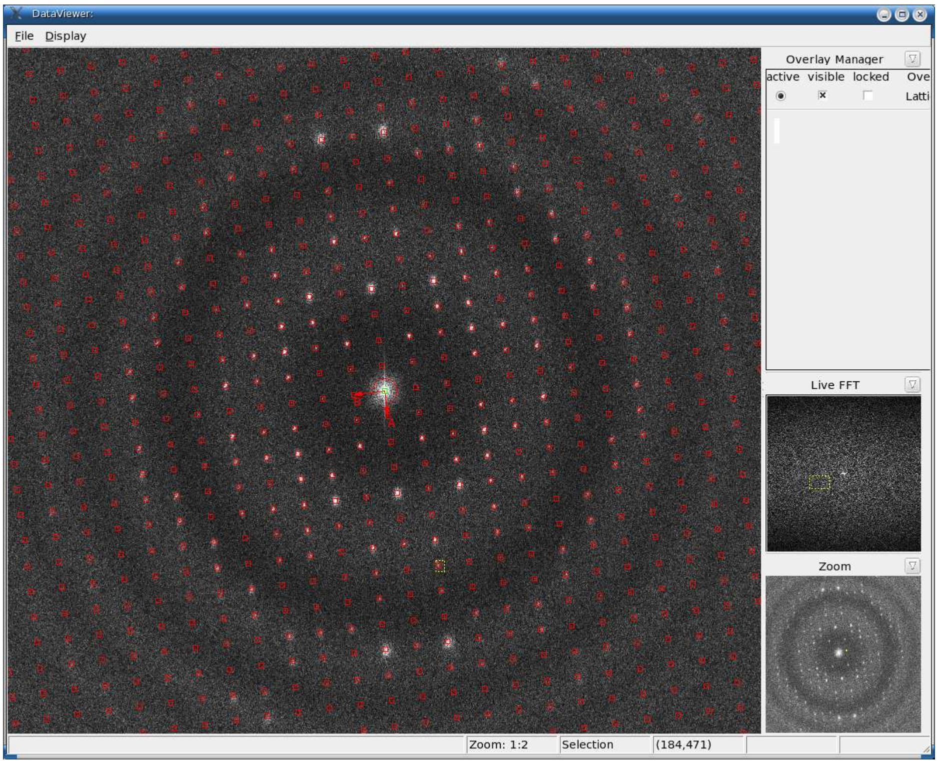


Figure 3.6: HasAR Indexing
The Fourier transform of an image of a HasA HasR crystal is indexed using the LatticeOverlay from IPLT. The Thon rings are clearly visible and there are visible diffraction spots up to a high resolution.

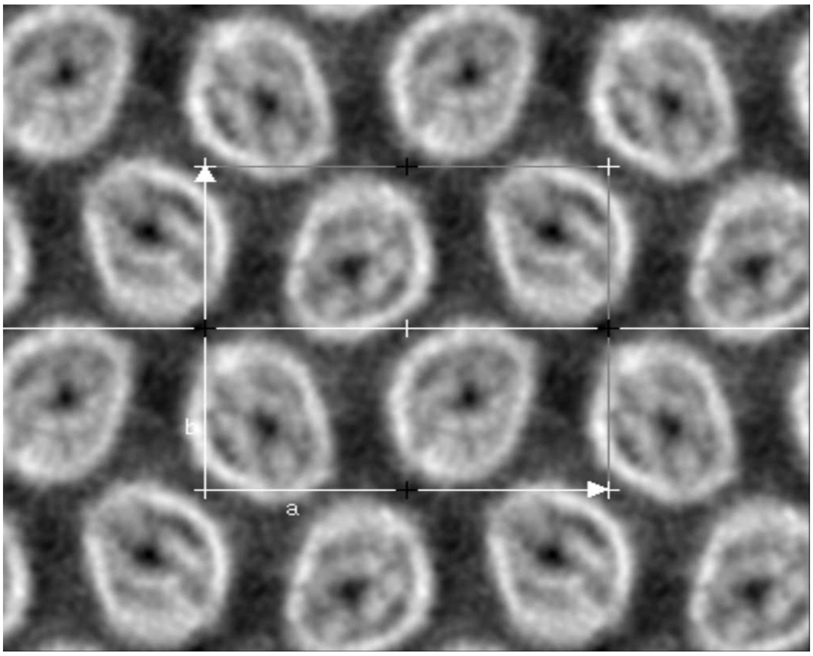
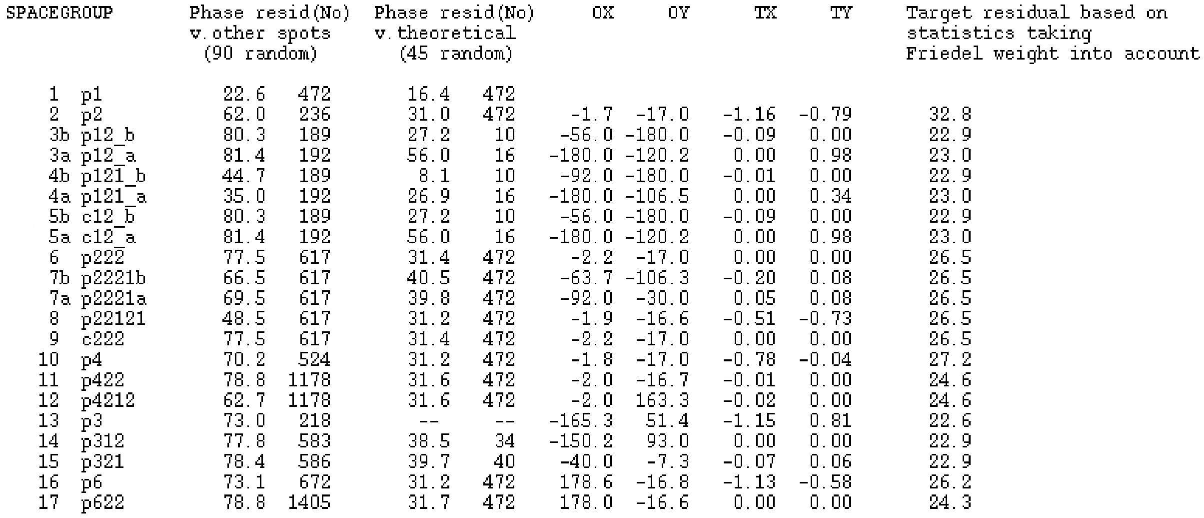


Figure 3.7: Unsymmetrized Projection Map of HasAR
This projection map result from the merging of 16 single images. No symmetry was used (p1). The unit cell has the parameters a = 126 Å, b = 101 Å and γ= 90°.

Projection Maps and Symmetry 16 images were used to calculate projection maps of HasAR. There was a strong indication judging from the unsymmetrized projection map (see figure3.7) that the protein monomers are arranged in a $p22_12_1$symmetry. Even if the symmetry calculations with the ALLSPACE program from MRC [9] from the single images did not clearly favor this symmetry (see figure 3.8), a projection map using this symmetry was calculated (see figure 3.9). The symmetrized and the unsymmetrized projections have a very high similarity. The differences between the single monomers in the unsymmetrized projection map are mainly in the details of the map. This observation is also supported by the AVRAMPHS statistics from MRC [9], which displays the phase error in relation with the resolution. The phase error is quite small up to a resolution of about 7 Å, and quite high at higher resolution. This could indicate that the crystal has a $p22_12_1$symmetry, where the single monomers are not perfectly in place. Another reason for this increase of the phase error could be a possible small error in the fitting of the CTF to the Thon rings. Since these rings are closer together at higher resolution, the possibility for phase errors of the reflections increases. The processing of the diffraction data will show what the main reason for the error increase at higher resolution is.

Interpretation of the Projection Map The calculated projection map allows first speculations on the arrangement of HasA and HasR. The crystal structure of HasA has been determined and is described in Arnoux et al. [2]. HasA has a size of 45×35×25 Å^3 and is composed of 4 α-helices and



```
SPACEGROUP        Phase resid(No)  Phase resid(No)     OX      OY      TX     TY      Target residual based on
                  v.other spots    v.theoretical                                      statistics taking
                   (90 random)       (45 random)                                      Friedel weight into account

    1  p1            22.6   472      16.4   472
    2  p2            62.0   236      31.0   472     -1.7  -17.0   -1.16  -0.79         32.8
   3b p12_b          80.3   189      27.2    10    -56.0 -180.0   -0.09   0.00         22.9
   3a p12_a          81.4   192      56.0    16   -180.0 -120.2    0.00   0.98         23.0
   4b p121_b         44.7   189       8.1    10    -92.0 -180.0   -0.01   0.00         22.9
   4a p121_a         35.0   192      26.9    16   -180.0 -106.5    0.00   0.34         23.0
   5b c12_b          80.3   189      27.2    10    -56.0 -180.0   -0.09   0.00         22.9
   5a c12_a          81.4   192      56.0    16   -180.0 -120.2    0.00   0.98         23.0
    6  p222          77.5   617      31.4   472     -2.2  -17.0    0.00   0.00         26.5
   7b p2221b         66.5   617      40.5   472    -63.7 -106.3   -0.20   0.08         26.5
   7a p2221a         69.5   617      39.8   472    -92.0  -30.0    0.05   0.08         26.5
    8  p22121        48.5   617      31.2   472     -1.9  -16.6   -0.51  -0.73         26.5
    9  c222          77.5   617      31.4   472     -2.2  -17.0    0.00   0.00         26.5
   10  p4            70.2   524      31.2   472     -1.8  -17.0   -0.78  -0.04         27.2
   11  p422          78.8  1178      31.6   472     -2.0  -16.7   -0.01   0.00         24.6
   12  p4212         62.7  1178      31.6   472     -2.0  163.3   -0.02   0.00         24.6
   13  p3            73.0   218       --     --   -165.3   51.4   -1.15   0.81         22.6
   14  p312          77.8   583      38.5    34   -150.2   93.0    0.00   0.00         22.9
   15  p321          78.4   586      39.7    40    -40.0   -7.3   -0.07   0.06         22.9
   16  p6            73.1   672      31.2   472    178.6  -16.8   -1.13  -0.58         26.2
   17  p622          78.8  1405      31.7   472    178.0  -16.6    0.00   0.00         24.3
```

Figure 3.8: ALLSPACE Statistics of one HasAR Image

The program ALLSPACE from MRC was used to determine the possible symmetries of the image. A low value in the *Phase resid(No) v. other spots* indicates a good fit for the corresponding symmetry. In this image one can see a value of 48.5 for the $p22_12_1$ symmetry, which indicates imperfections in this symmetry.

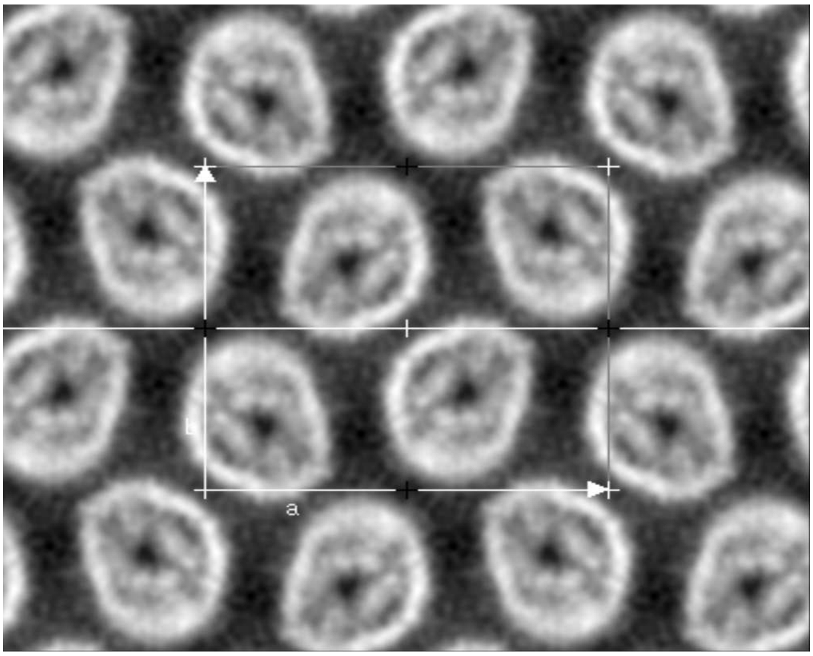
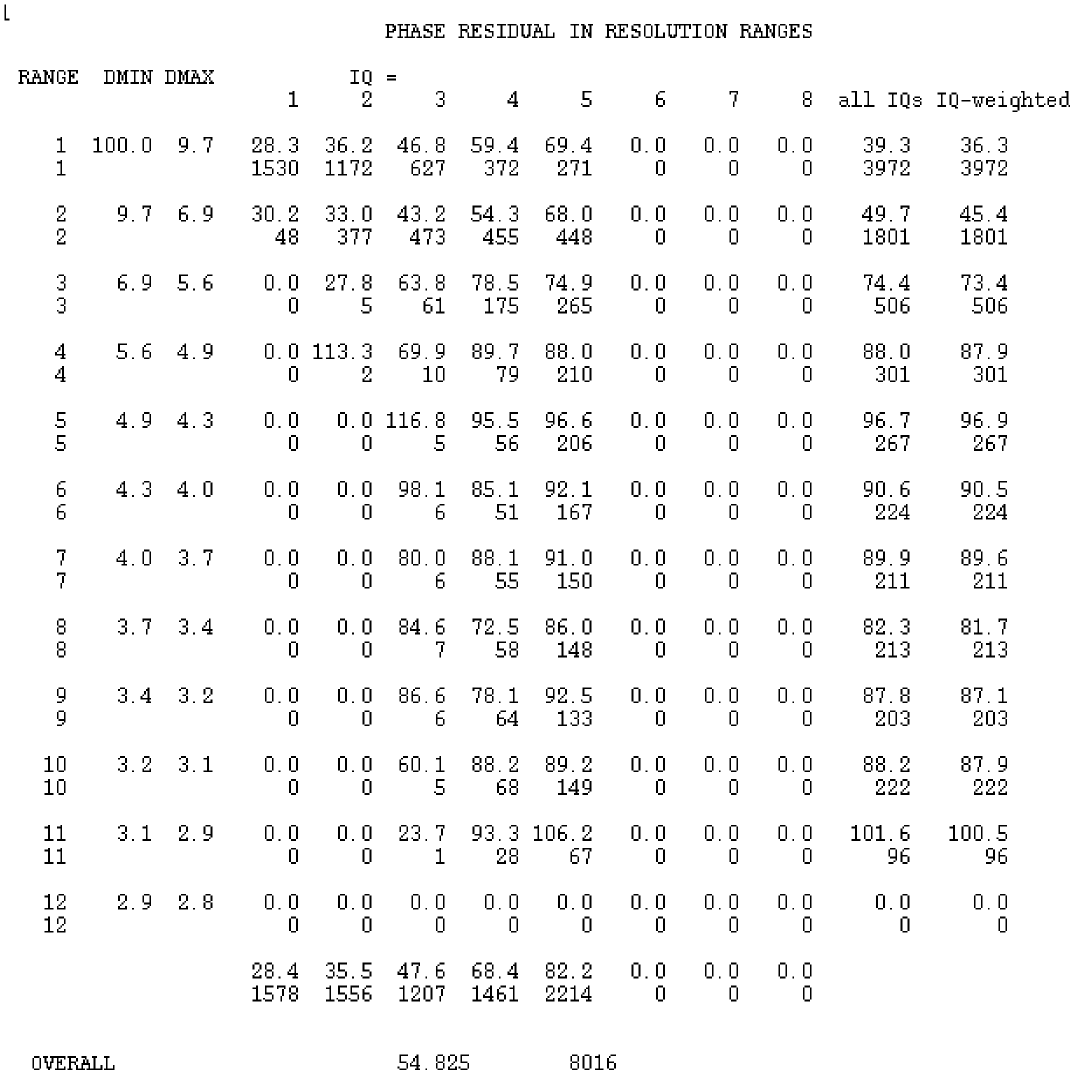


Figure 3.9: $p22_12_1$ Symmetrized Projection Map of HasAR

This projection map results from the same 16 images as the projection map presented in figure 3.7. The unit cell has the parameters a = 126 Å, b = 101 Å and γ= 90°.



```
                             PHASE RESIDUAL IN RESOLUTION RANGES

RANGE  DMIN DMAX          IQ =
                      1     2     3     4     5     6     7     8  all IQs IQ-weighted

   1  100.0   9.7  28.3  36.2  46.8  59.4  69.4   0.0   0.0   0.0    39.3     36.3
   1                1530  1172   627   372   271     0     0     0    3972     3972

   2    9.7   6.9  30.2  33.0  43.2  54.3  68.0   0.0   0.0   0.0    49.7     45.4
   2                  48   377   473   455   448     0     0     0    1801     1801

   3    6.9   5.6   0.0  27.8  63.8  78.5  74.9   0.0   0.0   0.0    74.4     73.4
   3                   0     5    61   175   265     0     0     0     506      506

   4    5.6   4.9   0.0 113.3  69.9  89.7  88.0   0.0   0.0   0.0    88.0     87.9
   4                   0     2    10    79   210     0     0     0     301      301

   5    4.9   4.3   0.0   0.0 116.8  95.5  96.6   0.0   0.0   0.0    96.7     96.9
   5                   0     0     5    56   206     0     0     0     267      267

   6    4.3   4.0   0.0   0.0  98.1  85.1  92.1   0.0   0.0   0.0    90.6     90.5
   6                   0     0     6    51   167     0     0     0     224      224

   7    4.0   3.7   0.0   0.0  80.0  88.1  91.0   0.0   0.0   0.0    89.9     89.6
   7                   0     0     6    55   150     0     0     0     211      211

   8    3.7   3.4   0.0   0.0  84.6  72.5  86.0   0.0   0.0   0.0    82.3     81.7
   8                   0     0     7    58   148     0     0     0     213      213

   9    3.4   3.2   0.0   0.0  86.6  78.1  92.5   0.0   0.0   0.0    87.8     87.1
   9                   0     0     6    64   133     0     0     0     203      203

  10    3.2   3.1   0.0   0.0  60.1  88.2  89.2   0.0   0.0   0.0    88.2     87.9
  10                   0     0     5    68   149     0     0     0     222      222

  11    3.1   2.9   0.0   0.0  23.7  93.3 106.2   0.0   0.0   0.0   101.6    100.5
  11                   0     0     1    28    67     0     0     0      96       96

  12    2.9   2.8   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0     0.0      0.0
  12                   0     0     0     0     0     0     0     0       0        0

                   28.4  35.5  47.6  68.4  82.2   0.0   0.0   0.0
                   1578  1556  1207  1461  2214     0     0     0


 OVERALL                        54.825        8016
```

Figure 3.10: AVRAMPHS Statistics of the $p22_12_1$ Symmetrized HasAR Projection

7 β-strands, arranged in a β-sheet. The inner densities visible in the monomers could represent the α-helices of HasA, since their spacial arrangement corresponds to the size of the HasA molecule. The outer ring of the monomer most probably represents the β-barrel structure proposed for HasR [27].

Outlook The HasAR project is now mainly continued processing the collected diffraction data. This is done by Ansgar Philippsen and Simon Bernèche. The processing of the diffraction data will show to what extent the $p22_12_1$symmetry assumption was sensible an whether the interpretation of the arrangement of HasA and HasR based on the projection map is correct.

The calculated projection of the HasAR complex can be used at a later time to restrict the search space for molecular replacement. This will be done when enough diffraction data is collected and processed.

3.7 Momentary Conclusions and Outlook

The production of 2D crystals, i.e. the expression, purification and crystallization of membrane proteins is one of the main focuses in our group. There are a lot of developments going on in this fields and I was part in some of them (mainly in the development of the cyclodextrin method and in the systematic screening of KdgM, KdgN and NanC).

The data collection with the electron microscope and the processing of such data in order to get information on the structure of these proteins is an other main focus in our group. The main effort in this field is put into the development of IPLT. The goal to make IPLT a flexible and powerful tool for image and diffraction processing is pursued with great energy. IPLT as it is now provides already a broad range of applications and much of the data processing from our crystals is done with IPLT. Still there is a lot of space for improvements. MRC is still heavily used and some problems like the lattice unbending need to be tackled in IPLT.

As IPLT is now in a state, where also general users, not familiar with the implementation details of IPLT, are using it to solve their problems, one can gain a lot of feedback about the users needs in terms of algorithms and GUI developments to implement.

Chapter 4

Conclusions and Outlook

4.1 Conclusions

The screening of conditions in order to produce 2D crystals suitable for structure determination is a very time consuming and unpredictable work. The space of possible conditions using different lipids and LPRs, detergents, salts and pH is huge and needs to be sampled over a fine mesh, since a small change in the condition can make a big difference in the result. From the gained experience in different projects aiming to produce 2D crystals, I learned that the only promising approach is to work in a systematic way and to keep an overview on the tested conditions, indicating the importance of an appropriate database.

The screening of the samples at the electron microscope needs to be done with great care. By looking carefully at the samples, a lot of information on the lipids, detergents or salts can be extracted from the experiments. It is difficult to define criteria to describe such observations quantitatively.

The development of IPLT in order to complement and replace existing software (such as MRC) proved to be sensible. The effort in developing a system having clear and comprehensible architecture allows us now to implement and test algorithms in an intuitive fashion. IPLT is now a software package suitable for various processing steps. The philosophy of extracting as much data as possible from the images and diffraction images instead of only processing the best available data, gives us insight on proteins whose crystals are not perfect and allows us to extract high resolution data from perfect crystals. The standardization of the processing finally made the data from different experiments comparable and reduced the time needed for a user to learn the software.

4.2 Outlook

As it is usual in the scientific world, all the described areas are under constant change and new ideas are evolving all the time. Here I just presented a small outlook on the projects I was/am involved with, and what advances or problems could follow.

Since a proper search for crystallization conditions includes a lot of work, an automation of some of the repetitive tasks is desired. The preparation of the conditions for dialysis or cyclodextrin experiments can be automated using commercially available devices. This has the advantage of preparing a high number of different starting mixtures in an accurate and less error prone way. The setup of the cyclodextrin experiments is also about to change completely, allowing the parallel processing of up to 96 conditions. This in-house built machine solves the mixing and the evaporation problems encountered with the old setup. In the effort on the automation of the sample screening in the frame of the HT3DEM collaboration one has to consider the difficulties in the automatic recognition and rating of the different objects that can be found on the grid. An experienced user can often better determine whether a region of the grid is worth zooming in or not, and has thus a smaller risk of overseeing interesting areas of the sample.

The automation of the whole process and the systematic search produces a lot of data on the tested conditions and the quality of these samples. A lab book is a good way to keep track of a certain number of conditions, but it makes it hard to compare results from different projects or over longer time periods. To ameliorate this situation a database for crystallization experiments is being implemented. The main work for this database was done by Christa Widerkehr and Thomas Braun, I contributed with my knowledge on the kind of data and the experimental setup. The database is designed to be flexible and expandable, but simple to use. The availability of the results of all persons performing crystallization experiments will help in gaining insight into the different factors influencing the formation of crystals. That way one can keep a better overview of these projects and avoiding double work.

IPLT is under constant development and as it has now gained a critical mass the usability will grow even faster. The implementations of the algorithms is easy and thus we can use our gained knowledge from theory (e.g. TCIF) or from practical experience to extract the highest amount of information from our data. The interoperability of IPLT with other software, the clear object oriented design and the platform independence are just beginning to be important aspects in the practical application of IPLT thus leaving a high potential for further developments.

4.3 Personal Conclusions

I have completed a degree in computer science with a specialisation in scientific computing and a minor in biochemistry at the ETH in Zurich. Having this strong theoretical background for the processing of scientific data, I wanted to gain more insight into the biological and physical processes producing scientific data and into the actual processing of such real data.

I have got the possibility to complete a PhD in the Engel group and could progress from the software development in the IPLT framework to crystallization experiments and method development in the laboratory environment and back to the image processing and software development with IPLT and MRC. This progress suited my talents and prior knowledge as well as the needs of the group.

IPLT Development My first steps in the Engel group were guided by Ansgar Philippsen and were made in an early stage of the IPLT development. Ansgar had already a *master plan* on how to develop a software for the processing of electron microscopy images and a good part of the basic frame was completed. IPLT was at that time by no means carved out of stone and so I could take part in design decisions and learn how the theoretical concepts can be implemented in a practical way. In the work on IPLT I could gain more practical experience in the development and in the implementation of algorithms.

Having no background in structural biology I learned a lot about concepts and techniques used in this field by attending the group seminars, taking lectures and by consulting the literature. The implementation of solutions for image processing and the discussions with all the group members gave me lots of insight into the processes generating the data we were supposed to process.

Growing Crystals To know the theoretical concepts and the software specific to structural biology was a good start. In order to get practical lab experience I took the opportunity to start wet-lab projects with Hervé Remigy. Hervé and other group members in the lab 300 introduced me to the work in the wet-lab and I could learn more about the technical and methodical aspects in the production of 2D protein crystals. My work in the lab was focused on the methodical and technical aspects of the production of 2D crystals, i.e. in the development of the new cyclodextrin technique for 2D crystallization and in the production and screening of crystals of three porins using the dialysis and the newly established cyclodextrin method. I spent a lot of time screening samples at the electron microscopes and learned to operate these instruments in an efficient and independent way. For educational purposes (Blockkurs) I learned how to operate the AFM, and I could also inspect the crystal arrangement of KdgM with this instrument.

Image Processing Once the protein crystals were available, it made sense to put again more energy in the computational side of structural biology. Having crystals of three proteins, I had plenty of material for data processing. First I finished the Linux port of the MRC programs so that everybody could work on a ordinary PC not being dependent anymore on the SGI machines. Further advances in IPLT, developed by the enlarged IPLT crew (Ansgar, Andi, Valerio, Simon, Johan and I), made it possible to use its new GUI components (overlays) for the determination of the initial parameters needed for the image processing with MRC.

The new possibilities in the user interaction was a great advantage in the processing of the difficult to handle image data coming from KdgM, NanC and KdgN. The gained expertise in the image processing made it possible for me to process the untilted images of HasAR. Since the crystals of HasAR were of a high quality, a projection map could be calculated in a relatively short time. During the time I was processing the data from my crystals, I was part of the IPLT development team and could deepen both my practical programming skills and my theoretical knowledge in image and diffraction processing.

Personal Outlook Coming from computer science with a theoretical background in scientific computing and biochemistry, I could profit a lot from the interdisciplinary team I was working with. It was interesting to see how people work and produce scientific data. Not only did I learn from them, but I hope I could also introduce some of my theoretical knowledge on how information should be structured into the lab. The techniques I learned here ranged from programming to pipeting and they build now a good background for my further scientific work: There are still a lot of interesting projects that need a solution and more challenges will develop from these.

Bibliography

[1] The CCP4 suite: programs for protein crystallography. *Acta Crystallogr D Biol Crystallogr*, 50(Pt 5):760–3, 1994.

[2] P. Arnoux, R. Haser, N. Izadi, A. Lecroisey, M. Delepierre, C. Wandersman, and M. Czjzek. The crystal structure of HasA, a hemophore secreted by Serratia marcescens. *Nat Struct Biol*, 6(6):516–20, 1999.

[3] H. R. Bigelow, D. S. Petrey, J. Liu, D. Przybylski, and B. Rost. Predicting transmembrane beta-barrels in proteomes. *Nucleic Acids Res*, 32(8):2566–77, 2004.

[4] N. Blot, C. Berrier, N. Hugouvieux-Cotte-Pattat, A. Ghazi, and G. Condemine. The oligogalacturonate-specific porin KdgM of Erwinia chrysanthemi belongs to a new porin family. *J Biol Chem*, 277(10):7936–44, 2002.

[5] G. Condemine, C. Berrier, J. Plumbridge, and A. Ghazi. Function and expression of an N-acetylneuraminic acid-inducible outer membrane channel in Escherichia coli. *J Bacteriol*, 187(6):1959–65, 2005.

[6] S. W. Cowan, R. M. Garavito, J. N. Jansonius, J. A. Jenkins, R. Karlsson, N. Konig, E. F. Pai, R. A. Pauptit, P. J. Rizkallah, J. P. Rosenbusch, and et al. The structure of OmpF porin in a tetragonal crystal form. *Structure*, 3(10):1041–50, 1995.

[7] S.W. Cowan, T. Schirmer, G. Rummel, M. Steiert, R. Ghosh, R.A. Pauptit, J.N. Jansonius, and J.P. Rosenbusch. Crystal structures explain functional properties of two E. coli porins. *Nature*, 358(6389):727–33, 1992.

[8] R. A. Crowther, L. A. Amos, J. T. Finch, D. J. De Rosier, and A. Klug. Three dimensional reconstructions of spherical viruses by fourier synthesis from electron micrographs. *Nature*, 226(5244):421–5, 1970.

[9] R. A. Crowther, R. Henderson, and J. M. Smith. MRC image processing programs. *J Struct Biol*, 116(1):9–16, 1996.

[10] B. L. de Groot, A. Engel, and H. Grubmuller. A refined structure of human aquaporin-1. *FEBS Lett*, 504(3):206–11, 2001.

[11] W.J. Degrip, J. Vanoostrum, and P.H. Bovee-Geurts. Selective detergent-extraction from mixed detergent/lipid/protein micelles, using cyclodextrin inclusion compounds: a novel generic approach for the preparation of proteoliposomes. *Biochem J*, 330 (Pt 2):667–74, 1998.

[12] D. J. DeRosier and P. B. Moore. Reconstruction of three-dimensional images from electron micrographs of structures with helical symmetry. *J Mol Biol*, 52(2):355–69, 1970.

[13] A. Engel, A. Hoenger, A. Hefti, C. Henn, R.C. Ford, J. Kistler, and M. Zulauf. Assembly of 2-D membrane protein crystals: dynamics, crystal order, and fidelity of structure analysis by electron microscopy. *J Struct Biol*, 109(3):219–34, 1992.

[14] J. Fiaux, E. B. Bertelsen, A. L. Horwich, and K. Wuthrich. NMR analysis of a 900K GroEL GroES complex. *Nature*, 418(6894):207–11, 2002.

[15] Y. Fujiyoshi, K. Mitsuoka, B.L. de Groot, A. Philippsen, H. Grubmuller, P. Agre, and A. Engel. Structure and function of water channels. *Curr Opin Struct Biol*, 12(4):509–15, 2002.

[16] J. M. Ghigo, S. Letoffe, and C. Wandersman. A new type of hemophore-dependent heme acquisition system of Serratia marcescens reconstituted in Escherichia coli. *J Bacteriol*, 179(11):3572–9, 1997.

[17] B. Gipson, X. Zeng, Z. Y. Zhang, and H. Stahlberg. 2dx-User-friendly image processing for 2D crystals. *J Struct Biol*, 2006.

[18] T. Gonen, Y. Cheng, P. Sliz, Y. Hiroaki, Y. Fujiyoshi, S. C. Harrison, and T. Walz. Lipid-protein interactions in double-layered two-dimensional AQP0 crystals. *Nature*, 438(7068):633–8, 2005.

[19] K. A. Graeme-Cook, G. May, E. Bremer, and C. F. Higgins. Osmotic regulation of porin expression: a role for DNA supercoiling. *Mol Microbiol*, 3(9):1287–94, 1989.

[20] P. J. Henderson and M. C. Maiden. Homologous sugar transport proteins in Escherichia coli and their relatives in both prokaryotes and eukaryotes. *Philos Trans R Soc Lond B Biol Sci*, 326(1236):391–410, 1990.

[21] R. Henderson and P. N. Unwin. Three-dimensional model of purple membrane obtained by electron microscopy. *Nature*, 257(5521):28–32, 1975.

[22] M. W. Heuzenroeder and P. Reeves. The tsx protein of Escherichia coli can act as a pore for amino acids. *J Bacteriol*, 147(3):1113–6, 1981.

[23] N. Izadi-Pruneyre, F. Huche, G. S. Lukat-Rodgers, A. Lecroisey, R. Gilli, K. R. Rodgers, C. Wandersman, and P. Delepelaire. The heme transfer from the soluble HasA hemophore to its

membrane-bound receptor HasR is driven by protein-protein interaction from a high to a lower affinity binding site. *J Biol Chem*, 281(35):25541–50, 2006.

[24] B. K. Jap. Molecular design of PhoE porin and its functional consequences. *J Mol Biol*, 205(2):407–19, 1989.

[25] T.C. Kaufmann, A. Engel, and H.W. Remigy. A novel method for detergent concentration determination. *Biophys J*, 90(1):310–7, 2006.

[26] W. Kukulski, A.D. Schenk, U. Johanson, T. Braun, B.L. de Groot, D. Fotiadis, P. Kjellbom, and A. Engel. The 5A structure of heterologously expressed plant aquaporin SoPIP2;1. *J Mol Biol*, 350(4):611–6, 2005.

[27] S. Letoffe, K. Wecker, M. Delepierre, P. Delepelaire, and C. Wandersman. Activities of the Serratia marcescens heme receptor HasR and isolated plug and beta-barrel domains: the beta-barrel forms a heme-specific channel. *J Bacteriol*, 187(13):4637–45, 2005.

[28] T. Ma, A. Frigeri, S. T. Tsai, J. M. Verbavatz, and A. S. Verkman. Localization and functional analysis of CHIP28k water channels in stably transfected Chinese hamster ovary cells. *J Biol Chem*, 268(30):22756–64, 1993.

[29] T. Ma, B. Yang, and A. S. Verkman. Cloning of a novel water and urea-permeable aquaporin from mouse expressed strongly in colon, placenta, liver, and heart. *Biochem Biophys Res Commun*, 240(2):324–8, 1997.

[30] A. J. Macpherson, M. C. Jones-Mortimer, P. Horne, and P. J. Henderson. Identification of the GalP galactose transport protein of Escherichia coli. *J Biol Chem*, 258(7):4390–6, 1983.

[31] C.A. Mannella. Phospholipase-induced crystallization of channels in mitochondrial outer membranes. *Science*, 224(4645):165–6, 1984.

[32] P. L. Martelli, P. Fariselli, A. Krogh, and R. Casadio. A sequence-profile-based HMM for predicting and discriminating beta barrel membrane proteins. *Bioinformatics*, 18 Suppl 1:S46–53, 2002.

[33] D. J. Muller and A. Engel. Voltage and pH-induced channel closure of porin OmpF visualized by atomic force microscopy. *J Mol Biol*, 285(4):1347–51, 1999.

[34] K. Murata, K. Mitsuoka, T. Hirai, T. Walz, P. Agre, J. B. Heymann, A. Engel, and Y. Fujiyoshi. Structural determinants of water permeation through aquaporin-1. *Nature*, 407(6804):599–605, 2000.

[35] T. Pellinen, H. Ahlfors, N. Blot, and G. Condemine. Topology of the Erwinia chrysanthemi oligogalacturonate porin KdgM. *Biochem J*, 372(Pt 2):329–34, 2003.

[36] A. Philippsen, W. Im, A. Engel, T. Schirmer, B. Roux, and D.J. Muller. Imaging the electrostatic potential of transmembrane channels: atomic probe microscopy of OmpF porin. *Biophys J*, 82(3):1667–76, 2002.

[37] A. Philippsen, A. D. Schenk, G. A. Signorell, V. Mariani, S. Berneche, and A. Engel. Collaborative EM image processing with the IPLT image processing library and toolbox. *J Struct Biol*, 2006.

[38] A. Philippsen, A.D. Schenk, H. Stahlberg, and A. Engel. Iplt–image processing library and toolkit for the electron microscopy community. *J Struct Biol*, 144(1-2):4–12, 2003.

[39] H.W. Remigy, D. Caujolle-Bert, K. Suda, A. Schenk, M. Chami, and A. Engel. Membrane protein reconstitution and crystallization by controlled dilution. *FEBS Lett*, 555(1):160–9, 2003.

[40] D. Ren, B. Navarro, H. Xu, L. Yue, Q. Shi, and D. E. Clapham. A prokaryotic voltage-gated sodium channel. *Science*, 294(5550):2372–5, 2001.

[41] W. Saenger, J. Jacob, K. Gessler, T. Steiner, D. Hoffmann, H. Sanbe, K. Koizumi, S. M. Smith, and T. Takaha. Structures of the Common Cyclodextrins and Their Larger Analogues-Beyond the Doughnut. *Chem Rev*, 98(5):1787–1802, 1998.

[42] T. Schirmer, T.A. Keller, Y.F. Wang, and J.P. Rosenbusch. Structural basis for sugar translocation through maltoporin channels at 3.1 A resolution. *Science*, 267(5197):512–4, 1995.

[43] T. Schirmer and P. S. Phale. Brownian dynamics simulation of ion flow through porin channels. *J Mol Biol*, 294(5):1159–67, 1999.

[44] G. E. Schulz. beta-Barrel membrane proteins. *Curr Opin Struct Biol*, 10(4):443–7, 2000.

[45] B. S. Schuwirth, M. A. Borovinskaya, C. W. Hau, W. Zhang, A. Vila-Sanjurjo, J. M. Holton, and J. H. Cate. Structures of the bacterial ribosome at 3.5 A resolution. *Science*, 310(5749):827–34, 2005.

[46] G. A. Signorell, T. C. Kaufmann, W. Kukulski, A. Engel, and H. W. Remigy. Controlled 2D crystallization of membrane proteins using methyl-beta-cyclodextrin. *J Struct Biol*, 2006.

[47] P. K. Smith, R. I. Krohn, G. T. Hermanson, A. K. Mallia, F. H. Gartner, M. D. Provenzano, E. K. Fujimoto, N. M. Goeke, B. J. Olson, and D. C. Klenk. Measurement of protein using bicinchoninic acid. *Anal Biochem*, 150(1):76–85, 1985.

[48] H. Sui, B. G. Han, J. K. Lee, P. Walian, and B. K. Jap. Structural basis of water-specific transport through the AQP1 water channel. *Nature*, 414(6866):872–8, 2001.

[49] K. Takata, T. Matsuzaki, and Y. Tajika. Aquaporins: water channel proteins of the cell membrane. *Prog Histochem Cytochem*, 39(1):1–83, 2004.

[50] S. Tornroth-Horsefield, Y. Wang, K. Hedfalk, U. Johanson, M. Karlsson, E. Tajkhorshid, R. Neutze, and P. Kjellbom. Structural mechanism of plant aquaporin gating. *Nature*, 439(7077):688–94, 2006.

[51] J. M. Verbavatz, D. Brown, I. Sabolic, G. Valenti, D. A. Ausiello, A. N. Van Hoek, T. Ma, and A. S. Verkman. Tetrameric assembly of CHIP28 water channels in liposomes and cell membranes: a freeze-fracture study. *J Cell Biol*, 123(3):605–18, 1993.

[52] T. Walz, B. L. Smith, P. Agre, and A. Engel. The three-dimensional structure of human erythrocyte aquaporin CHIP. *EMBO J*, 13(13):2985–93, 1994.

[53] T. Walz, B. L. Smith, M. L. Zeidel, A. Engel, and P. Agre. Biologically active two-dimensional crystals of aquaporin CHIP. *J Biol Chem*, 269(3):1583–6, 1994.

[54] C. Wandersman and P. Delepelaire. Bacterial iron sources: from siderophores to hemophores. *Annu Rev Microbiol*, 58:611–47, 2004.

[55] J. Ye and B. van den Berg. Crystal structure of the bacterial nucleoside transporter Tsx. *EMBO J*, 23(16):3187–95, 2004.

[56] O. Yildiz, K. R. Vinothkumar, P. Goswami, and W. Kuhlbrandt. Structure of the monomeric outer-membrane porin OmpG in the open and closed conformation. *EMBO J*, 25(21):5240, 2006.

Appendix A

Abbreviations

- AFM atomic force microscope
- AQP aquaporin
- BCA bicinchoninic acid
- CTF Contrast transfer function
- DDM n-Dodecyl β-D-maltoside
- DMPC 1,2Dimyristoyl-snGlycero3Phosphocholine
- DOPG dioleoylphosphatidylglycerol
- DOPS 1,2-Dioleoyl-sn-Glycero-3-[Phospho-L-Serine]
- DOPC dioleoyl-phosphatidylcholine
- DTT dithiothreitol
- EM electron microscope
- GUI graphical user interface
- HMM hidden markov model
- IPLT **i**mage **p**rocessing **l**ibrary and **t**oolbox
- LDAO Lauryldimethyloxide
- LPR lipid to protein ratio
- MBCD methyl-β-cyclodextrin

- MRC Laboratory of Molecular Biology, Cambridge - Image processing software developed there
- NMR nuclear magnetic resonance
- NN Neuronal Network
- OG Octyl Glucoside
- ORF Open Reading Frame
- SGI silicon graphic
- TCIF tilted contrast-imaging function
- TX-100 Triton X-100

Profile

Gian Andrea Signorell

Binngerstrasse 15
CH - 4051 Basel
Tel. +41 79 401 76 28
giani.signorell@unibas.ch

Personal Data:

Date of Birth	16/12/1978
Nationality	Swiss
Marital Status	Married
Languages	German, Romansh, English, French, Italian, Swedish

EDUCATION AND QUALIFICATIONS

2003 to the present	University of Basel, MIH, Group Engel PhD student in structural biology Lectures including product management, marketing and drug development
1998 - 2003	Swiss Federal Institute of Technology Zurich (ETH) Graduate engineer in computer science with a minor in biochemistry
2002 - 2003	KTH Stockholm Erasmus semester as part of the graduate program in computer science
1993 - 1998	Lyceum Alpinum Zuoz Matura Typus C (high school degree in natural sciences and mathematics)

WORK EXPERIENCE

2003 to the present	PhD student, University of Basel Development of image-processing software (iplt.org) Production and analysis of 2D protein crystals Electron microscopy, atomic force microscopy and image processing
2001 - 2002	Scientific assistant at ETH Zurich Assisting exercises in scientific computing Programming learning software for the exercises
2001	Internship at Hoffman La Roche Ltd. Work in the departments of bioinformatics and structural chemistry
2001	Freelancer, Rapp Group Consulting and implementation work for the LSVA homepage
1996 - 2003	Other jobs have included ski teaching and web-design

LANGUAGE SKILLS

German	native language
Romansh	native language
English	primary working language
French	working language – very good oral skills
Italian	good oral skills
Swedish	basic skills in spoken and written

IT SKILLS

Operating Systems	Windows, Unix/Linux, and MacOS
Programming Experience	C++, C, FORTRAN, JAVA, Python
Image Processing	MRC, IPLT, Digital Micrograph, PhotoShop, GIMP
Other Software	MS-Office, LaTeX, web design tools

OTHER QUALIFICATIONS

2002	Didactic training for tutors, ETH Zurich
2002	Training as J+S instructor (Alpine skiing)
1996	Training as children skiing instructor

TIME SPENT ABROAD

2004	1 month Madagascar and Mauritius
2002-2003	6 Months Stockholm with the ERASMUS exchange program
2001	2 months Thailand and Vietnam
1999	1 month Mexico
1998	3 months Australia

Other travels to European countries and Israel.

OTHER ACTIVITIES

2006 to the present	Member of the executive board ETH Alumni, GEP Basel
Sports	Fencing, skiing, basketball, unihockey and diving
Leisure	Travelling, cooking and reading

DOCUMENTS ATTACHED

Publications
References
Certifications

Basel, June 2007

Gian Andrea Signorell

PUBLICATIONS

Submitted	Signorell, G.A., Schenk, A.D., Hebert, J., Mariani, V., Engel, A., Philippsen, A. 05/2007. Versatile Image Processinf of Electron Crystallography Data in IPLT.
Submitted	Signorell, G.A., Remigy, H.W., Chami, M., Condemine, G., Schenk, A.D., Philippsen, A., Engel, A. 05/2007. Projection Maps of Three Members of the KdgM Outer Membrane Protein Family.
2006	Signorell, G.A., Kaufmann, T.C., Kukulski, W., Engel, A. And Remigy, H.W., 2006. Controlled 2D crystallization of membrane proteins using methyl-beta-cyclodextrin. J Struct Biol 157:321-328.
2006	Philippsen, A., Schenk, A.D., Signorell, G.A., Mariani, V., Berneche S. and Engel, A., 2006. Collaborative EM image processing with the IPLT image processing library and toolbox. J Struct Biol 157: 28-37.

REFERENCES

Prof. Andreas Engel
Responsible Professor of PhD thesis

University of Basel, MIH
Klingelbergstrasse 50-70
4056 Basel
Tel. +41 61 267 22 62
Email: andreas.engel@unibas.ch

Dr. Ansgar Philippsen
Supervisor of PhD thesis

University of Basel, MIH
Klingelbergstrasse 50-70
4056 Basel
Tel. +41 61 267 14 76
Email: ansgar.philippsen@unibas.ch